

ANALYSIS OF TEMPORAL AND SPATIAL TRENDS OF GREENHOUSE GAS
EMISSION SOURCES WITH CONDITIONAL BIVARIATE
PROBABILITY FUNCTIONS

by

Megan Katherine Walsh

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Civil and Environmental Engineering

The University of Utah

May 2017

Copyright © Megan Katherine Walsh 2017

All Rights Reserved

The University of Utah Graduate School

STATEMENT OF THESIS APPROVAL

The thesis of **Megan Katherine Walsh**
has been approved by the following supervisory committee members:

Brian McPherson , Chair **3/7/2017**
Date Approved

Simon Brewer , Member **3/8/2017**
Date Approved

Michael Barber , Member **3/8/2017**
Date Approved

and by **Michael Barber** , Chair/Dean of
the Department/College/School of **Civil and Environmental Engineering**

and by David B. Kieda, Dean of The Graduate School.

ABSTRACT

The primary purpose of this study is to develop a method for estimating the location of a single point source of a greenhouse gas (GHG) leak. The case study is the University of Utah campus. Specifically, we hypothesized that sewer access covers (“manholes”) are significant sources of GHG emissions on campus, and we used these known point source locations to test the ability of standard GHG measurement instruments to develop methods for identifying a priori unknown locations. We sited gas analyzers at specific distances and directions from sewer access covers and then evaluated whether tailored probability density functions would “point” towards the sites.

All GHG data were measured by a Picarro[®] cavity ring-down spectrometer (CRDS) gas analyzer and contemporaneous wind speed and direction were measured with a 3-dimensional (3D) anemometer. Since this study focused on locating leakage sources, we assigned the concentration threshold for leakage concentration to be the 99th percentile concentration of each dataset, a common approach in leakage detection studies. The leakage concentration data along with corresponding wind speed and direction were used to create conditional bivariate probability function (CBPF) plots. All CBPF plots were constructed for varying time spans throughout the day at each collection site to discern the probable location of GHG leakage sources. The results revealed that all 99th percentile concentrations were associated with lower wind speeds (<1.5 m/s). Higher

GHG concentrations associated with high wind speeds were most likely diluted before the signal could reach the receptor. Furthermore, the study site is characterized by hilly terrain, large buildings, and a moderate amount of large vegetation (trees) that likely tend to disperse what would otherwise be detectable GHG signals. Differences in CBPF plots at different receptor (tower) locations confirm that the distance between the tower and leakage sources is a critical issue. To our knowledge, this study is the first to apply CBPF's with the intent of quantifying trends in spatio-temporal GHG distributions from point leakage sources and determining probable locations of those sources.

TABLE OF CONTENTS

ABSTRACT	iii
LIST OF TABLES	vii
LIST OF FIGURES	viii
ACKNOWLEDGEMENTS	x
Chapters	
1. INTRODUCTION	1
2. METHOD	4
2.1 Study Area	4
2.2 Data	5
2.3 CPF/CBPF Formulation.....	6
3. RESULTS/DISCUSSION	12
3.1 TL1 Diagrams	12
3.2 FASB1 Diagrams	14
4. CONCLUSIONS	27
5. FUTURE DIRECTIONS	30
Appendices	
A: R SCRIPTS FOR DATA ANALYSIS ON UNIVERSITY OF UTAH CAMPUS (TL1 AND FASB)	32
B: UNIVERSITY OF UTAH UNION ROOF CASE STUDY METHODOLOGY	79
C: TIME SERIES PLOTS CREATED FOR DATA CLEANING PURPOSES.....	98

D: R SCRIPTS FOR DATA ANALYSIS ON UNIVERSITY OF UTAH CAMPUS UNION BUILDING FOR OCTOBER 2016	109
REFERENCES	123

LIST OF TABLES

- 2.1 Tower equipped with Picarro[®] instruments data collection periods (*indicates flux chamber data collected concurrently at sewer access covers.....11
- 2.2 Descriptive statistics of entire dataset. Note that the FASB1 March 2014 data are a composite dataset of three consecutive daytime data collection periods. TL1 and FASB1 2013 were overnight data collection periods.....11

LIST OF FIGURES

2.1	Site area.....	9
2.2	TL1 time series.....	10
2.3	FASB1 time series	10
3.1	TL1 daytime and nighttime wind roses	17
3.2	TL1 CBPF plots. (a) 99 th perc. CO ₂ during the day, (b) 99 th perc. CH ₄ during the day, (c) 99 th perc. CO ₂ during the night, (d) 99 th perc. CH ₄ during the night.....	18
3.3	TL1 wind roses. (a) AM rush hour wind rose, (b) PM rush hour wind rose. AM rush hour was determined to be between 7am and 10am in the morning. PM rush hour was set as 4pm to 6pm in the evening.....	19
3.4	TL1 CBPF plots. (a) 99 th perc. CO ₂ during AM rush hour, (b) 99 th perc. CH ₄ during AM rush hour, (c) 99 th perc. CO ₂ during PM rush hour, (d) 99 th perc. CH ₄ during PM rush hour.....	20
3.5	TL1 early morning wind rose.....	21
3.6	TL1 early morning (4am to 7am) CBPF plots. (a) 99 th perc. CO ₂ , (b) 99 th perc. CH ₄	21
3.7	FASB1 daytime and nighttime wind roses.....	22
3.8	FASB1 CBPF plots. (a) 99 th perc. CO ₂ during the day, (b) 99 th perc. CH ₄ during the day, (c) 99 th perc. CO ₂ during the night, (d) 99 th perc. CH ₄ during the night.....	23
3.9	FASB1 rush hour wind rose.....	24
3.10	FASB1 evening rush hour CBPF plots. (a) 99 th perc. CO ₂ , (b) 99 th perc. CH ₄	24
3.11	FASB1 early morning wind rose.....	25
3.12	TL1 early morning CBPF plots. (a) 99 th perc. CO ₂ , (b) 99 th perc. CH ₄	25

3.13	Wind rose diagram for data collected in March 2014 at FASB1.....	26
3.14	CBPFs for data collected at FASB1 during March 2014. (a) 99th perc. CO ₂ , (b) 99th perc. CH ₄	26

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Dr. McPherson, and my committee members, Dr. Brewer and Dr. Barber. Their feedback and support helped me tremendously. I would also like to thank everyone in the Carbon Science & Engineering Group (CSER), specifically Rich Esser and Robert Veljak, for providing so much help with the Licor tower. I would also like to thank the Ray Olpin Union Building Administration Office for allowing us to set up a research tower on the roof of the Union building. This work wouldn't have been possible without former CSER students Lindsay Minck, Amanda Fenton, and Kevin McCormack, whose work paved the way for my thesis, especially Lindsay who collected the data that were used in this analysis. I would also like to thank the University of Utah Department of Civil & Environmental Engineering, the Energy & Geoscience Institute, and the U.S. Department of Energy for funding through award no. DE-FC26-05NT42591.

CHAPTER 1

INTRODUCTION

Visual tools are commonly used with air pollution concentration data and meteorological data to ascertain likely directions of the source of the emissions. Wind rose diagrams, circular histograms whose bins correspond to azimuthal directions, are often used to depict wind direction data (Fisher, 1996). Pollutant concentration rose diagrams use concentration data and corresponding wind direction data to determine the likely direction of a pollutant source (Cosemans et al., 2008; Ruiz et al., 2014), but pollutant concentration roses often rely on low temporal resolution concentration and meteorological data, impeding ability to locate shorter duration emissions difficult. Furthermore, pollutant concentration roses often rely on the average pollutant concentration in a wind direction bin (range of degrees, such as 0 to 30 degrees), and thus isolated high concentration spikes can go undetected when using the average pollutant concentration.

Conditional probability functions (CPF) are another approach for identifying the possible location(s) of a pollutant source. The concept of a CPF was first published by Ashbaugh et al. (1985) who used it to determine the source direction of pollutants. In graphical form, CPF diagrams are polar plots that depict the probability that concentration exceeds a predetermined threshold in a given wind direction. Kim and

Hopke (2004) found CPFs to be a simple tool to determine possible directions of $PM_{2.5}$ sources with hourly data collected at sites around the United States. Pekney et al. (2006) used 24-hour averaged $PM_{2.5}$ concentration data with 15-minute wind data from Pittsburgh, PA to create CPF plots and found that CPFs results were helpful in determining local point sources of $PM_{2.5}$. Graphical CPFs used by Xie and Berkowitz (2007) demonstrated the direction from which different categories of volatile organic compounds (VOC) originated in the Houston, TX area.

A conditional bivariate polar diagram (CBPF) combines a CPF and a bivariate polar plot, typically with a third variable plotted on the radial axis (Uria-Tellaetxe and Carslaw, 2014; Carslaw, 2015). In most cases, the third variable is wind speed. Uria-Tellaetxe and Carslaw (2014) used 15-min SO_2 , PM_{10} , NO_x , and wind data to demonstrate the source characterization ability of CBPFs when different ranges of concentration are plotted (as opposed to specifying a specific threshold). Krecl et al. (2016) used CBPFs to plot different concentration ranges at different time periods throughout the day, revealing two different sources of black carbon: diesel bus emissions during the day and biomass burning at night.

To detect and quantify CO_2 and CH_4 emissions from sewer access points on the University of Utah campus, soil flux chambers were used in conjunction with a single tower equipped with a three-dimensional (3D) anemometer and a Picarro[®] cavity ring-down spectrometer (CRDS) gas analyzer during two previous and separate studies (Varland, 2014; Minck, 2015). Both studies hypothesized that sewer access points were significant point sources of greenhouse gas (GHG) emissions on campus. The soil flux chamber study results indicated that the annual carbon emission from the 11 sewer access

points measured was 1.066 Metric Tons CO₂ equivalent per year (MtCO₂e), only a fraction of the average carbon emissions associated with an individual US citizen (on the order of 10 to 20 tons per year). The preliminary results from the tower study were inconclusive as to identifying the location of CO₂ and CH₄ emission sources based solely on tower measurements. Thus, based on these previous studies and other published literature, a major goal of this thesis was to demonstrate how CBPFs can be used to explore spatial and temporal trends of probable GHG source locations in an urban setting. Applying CBPFs to identify source directions corresponding to high concentration levels during specific time spans may allow researchers to discern probable source locations. To our knowledge, this paper is the first to demonstrate how CPFs or CBPFs can be used with concentration data of GHGs such as CO₂ and CH₄.

CHAPTER 2

METHOD

2.1 Study Area

Initially, two distinct experiments were conducted to characterize and analyze sewer gas emissions from access covers on the University of Utah campus. The first (Varland, 2014) focused on measuring CO₂ and CH₄ flux from sewer access covers by employing a custom-made flux chamber design. The second experiment (Minck, 2015) consisted of placing a 3D sonic anemometer and the inlet tube for the laser-based Picarro[®] CRDS on a ten-foot-tall tower. The tower setup collected CO₂, CH₄, wind direction, and wind speed data at a frequency of ten times per second (10 Hz).

From September 2013 to March 2014, the tower was placed at two locations (TL1 and FASB1) chosen for their proximity to a sewer access point that was determined by Varland (2014) to have high CO₂ and CH₄ emissions (see Figure 2.1 for study location). Measurements were made when the equipment could be supervised to avoid possible damage or data interruption, as it was deployed on the University of Utah campus. This led mostly to data collection periods of several hours, but also a few coordinated overnight and full-day data sets. See Table 2.1 for data collection session information associated with the tower.

2.2 Data

The data used for the construction of CBPFs in this study consisted of two overnight data collection periods at sites TL1 and FASB1 (Figure 2.1) in October, 2013 and three consecutive daytime collection periods at FASB1 in March, 2014. The overnight TL1 and FASB1 datasets were analyzed separately. The three consecutive daytime collection periods (March, 2014) were combined into one dataset since the data were collected at the same location from approximately 11 am to 5 pm on three consecutive days.

The 3D anemometer and Picarro[®] CRDS enabled high frequency data collection. Wind speed, wind direction, and GHG concentration values were collected at 10 Hz frequency. These 10 Hz data provide higher temporal resolution than previous studies that utilized either 15-minute, hourly, or daily time intervals to produce CPFs and CBPFs (Kim and Hopke, 2004; Pekney et al., 2006; Uria-Tellaetxe and Carslaw, 2014). Descriptive statistics were calculated (see Table 2.2). Time series of CO₂, CH₄, and wind speed for the overnight data collection sessions were plotted to evaluate for diurnal trends in concentration or wind speed, as well as help determine any specific sub-periods that should be examined using the CBPFs (Figures 2.2 and 2.3).

The data were partitioned into daytime and nighttime periods as well as different time spans for temporal analysis. In addition to sewer access covers (“manholes”), we hypothesized that emissions from local traffic may also be detected. Thus, data collected during both morning and afternoon rush hour were analyzed specifically to determine whether significant differences in the probable source locations were indicated by the CBPFs. Morning rush hour was assumed to occur between 7 am and 10 am in the morning; evening rush hour period was assumed to occur from 4 pm to 6 pm. In addition

to rush hour commuting periods, the time period from 4 am to 7 am was also isolated for analysis because peaks in concentration were observed during these times within multiple datasets. The peaks in CO₂ and CH₄ concentration were attributed to campus maintenance operations starting for the day and buildings initiating their heating systems. See Appendix A for R scripts.

2.3 CPF/CBPF Formulation

Ashbaugh et al. (1985) initially developed a CPF method to determine the probable locations of pollutant emission sources with respect to a receptor. This method was tested using data collected at Grand Canyon National Park. Ashbaugh et al. (1985) defined CPF as:

$$CPF_{\Delta\theta} = \frac{m_{\Delta\theta|c \geq x}}{n_{\Delta\theta}} \quad (2.1)$$

where $m_{\Delta\theta}$ is the number of samples in a specified wind sector $\Delta\theta$, that exceeds a predetermined concentration threshold, x . $n_{\Delta\theta}$ is the total number of samples regardless of concentration, in the wind sector $\Delta\theta$. Typically, the size of a wind sector θ is a bin of degrees around a circle, such as 5 or 10 degrees. CPF values are plotted on polar plots corresponding to azimuthal directions of wind direction. The probability that a pollutant concentration in a certain direction exceeded a predetermined threshold is plotted on the radial axis. Visually, the probability values are the radial component in a polar graph wherein the wind direction directly corresponds to the azimuthal direction.

The concentration threshold can be an assigned value or a percentile that is determined on the sample's concentration ranges, such as a 75th percentile or 90th percentile. For this study, we elected to use the 99th percentile of each dataset because the high, dataset-specific value represents a concentration peak distinct to each dataset (Minck, 2015). A further motivation for our selection of the 99th percentile is that in leakage source identification studies, high percentiles such as 95th and 99th percentiles are often examined because they facilitate ability to distinguish likely background concentrations from those that are representative of a leak (Lewicki et al., 2009; Yang et al., 2011).

Uria-Tellaetxe and Carslaw (2014) extended the CPF method and created the CBPF, a bivariate diagram incorporating a third variable in addition to concentration and azimuth. Visually the CPF and CBPF are different because the conditional probability is represented by a color ramp in the CBPF and the radial component represents the third variable (in this case, wind speed). The CBPF is mathematically defined as:

$$CBPF_{\Delta\theta} = \frac{m_{\Delta\theta, \Delta\mu | c \geq x}}{n_{\Delta\theta, \Delta\mu}} \quad (2.2)$$

where $m_{\Delta\theta, \Delta\mu}$ is the number of samples in a specified wind sector $\Delta\theta$ and with wind speed $\Delta\mu$, that exceeds a predetermined concentration threshold, x . $n_{\Delta\theta, \Delta\mu}$ is the total number of samples in wind sector $\Delta\theta$ with wind speed $\Delta\mu$.

The CBPFs were plotted using the openair package (Carslaw, 2015) in R. Generalized additive models (GAMs) are used to fit the concentration surface with hourly mean wind speed and wind direction data. Fitting a GAM to the data reduces the amount of noise

plotted, resulting in a clearer rendition of concentration trends. Data points associated with all wind speeds were retained for this analysis (Carslaw, 2015). In some previous studies, data points associated with exceedingly large or very small wind speeds were considered outliers and therefore extracted from the analysis depending on the goal of the study and site setup (Kim and Hopke, 2004; Pekney et al., 2006). See Appendix A for scripts used to create the CBPF plots for each dataset.

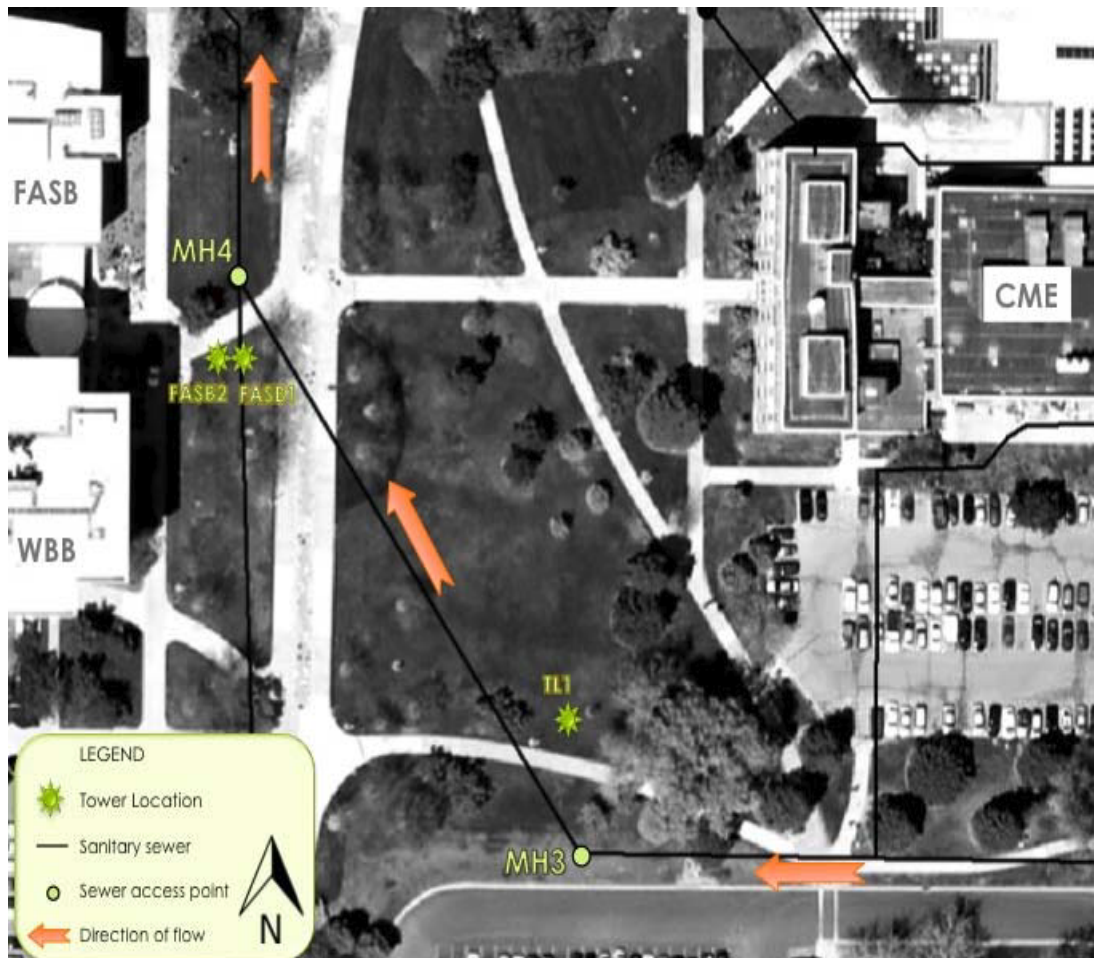


Figure 2.1 Site area (Source: Minck, L. K. Design, Deployment, and Initial Calibration of a Tower for Greenhouse Gas Measurements, University of Utah, 2015. Copyright © Lindsay K. Minck 2015)

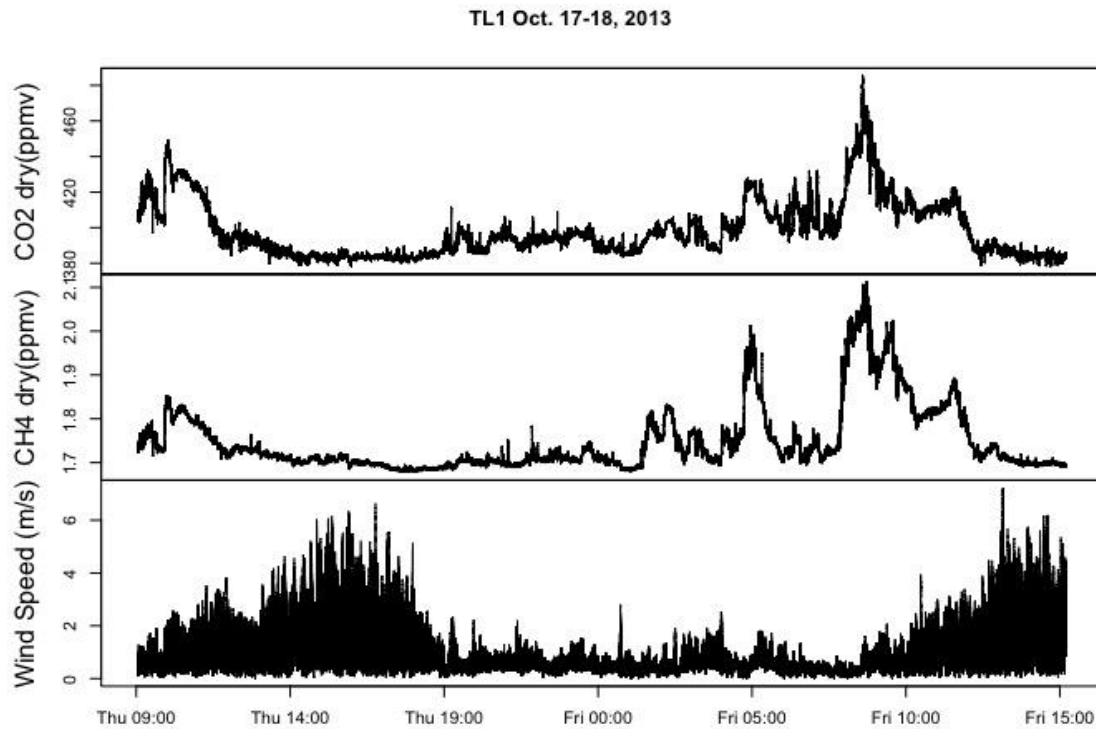


Figure 2.2 TL1 time series

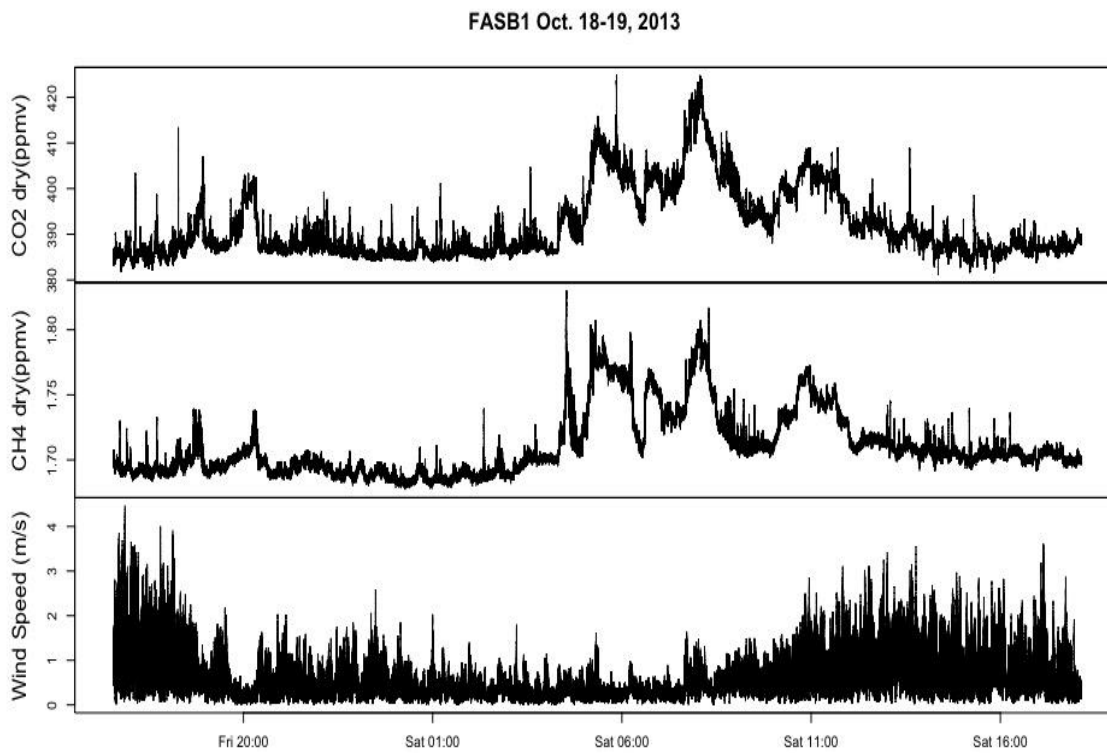


Figure 2.3 FASB1 time series

Table 2.1 Tower equipped with Picarro[®] instruments data collection periods (*indicates flux chamber data collected concurrently at sewer access covers)

Site	Start	End	Duration
TL1	2013-10-17 09:04:00	2013-10-18 15:11:21	30H 7M 22S
FASB1	2013-10-18 16:34:02	2013-10-19 18:08:12	25H 34M 11S
FASB1*	2014-03-06 12:30:41	2014-03-06 16:57:06	4H 26M 24S
FASB1*	2014-03-07 10:56:38	2014-03-07 17:02:42	6H 6M 3S
FASB1*	2014-03-08 11:30:31	2014-03-08 17:34:58	6H 4M 26S

Table 2.2 Descriptive statistics of entire dataset. Note that the FASB1 March 2014 data are a composite dataset of three consecutive daytime data collection periods. TL1 and FASB1 2013 were overnight data collection periods.

	TL1			FASB1 2013			FASB1 2014 (daytime)		
	CO ₂	CH ₄	WS	CO ₂	CH ₄	WS	CO ₂	CH ₄	WS
Min	378.20	1.68	0.00	381.20	1.68	0.00	386.50	1.69	0.00
Mean	398.70	1.75	0.90	392.00	1.71	0.56	394.90	1.72	1.66
Median	394.30	1.72	0.69	388.40	1.70	0.46	394.30	1.71	1.44
Max	485.40	2.11	7.23	424.80	1.83	4.47	456.80	2.17	9.62
99th perc.	450.07	2.03	3.28	416.17	1.79	1.92	416.11	1.76	4.87
Stand. Dev.	15.73	0.08	0.67	7.64	0.03	0.40	5.44	0.02	1.02
# of Obs	3,679,055			3,133,208			2,034,060		

CHAPTER 3

RESULTS/DISCUSSION

The resulting wind roses, CBPF plots, and analysis are grouped below for each data collection period (TL1, FASB1, FASB1-March 2014).

3.1 TL1 Diagrams

The daytime and nighttime wind roses (Figure 3.1) suggest that the predominant wind direction shifts from the northwest during the day to the northeast during night. The CBPF data for TL1 (Figure 3.2) are subject to impacts from diurnal mountain wind systems because the site is so close to a canyon mouth. The diurnal mountain wind system is thermally driven; early in the morning winds are driven “upslope” by thermal differences and the trend reverses in during nighttime when “downslope” winds occur (Zardi and Whiteman, 2013). Daytime CO₂ and CH₄ CBPFs at the 99th percentile (Figures 3.2a, 3.2b) both suggest that an emission source is located to the northwest of the tower location. The data shown in Figure 3.2 suggest that this source could be either a sewer access cover (MH4) or the main roadway that borders the north side of campus (100 South/North Campus Drive). The source is associated with low wind speeds, approximately 1 m/s, suggesting it is relatively nearby and reaches the tower before it is subject to significant dispersion. The nighttime CBPFs at the 99th percentile (Figures 3.2c

and 3.2d) suggest two CO₂ and CH₄ emission sources both to the southeast and the southwest. The University of Utah Union building and parking lots are south of the TL1 site.

Rush hour wind roses (Figure 3.3) show that the dominant wind directions for the morning and evening rush hour periods were very similar. Wind speeds were slightly greater during the PM rush hour. Both CO₂ and CH₄ evening rush hour CBPFs (Figure 3.4) exhibit 99th percentile concentrations associated with greater wind speeds (~ 2 to 4 m/s) compared to the morning rush hour plots (~ 0.5 to 1 m/s). The morning and evening rush hour CBPFs of CO₂ suggest that the source is coming from the north where the main roadway, 100 S./North Campus Dr., is located. 100 S./North Campus Dr. is not only congested during rush hour with cars in this area, but there is also a busy bus stop located at the bend in the road, just north of the FASB1 location. The 99th percentile value of CO₂ from the morning rush hour (471 ppm) and the evening rush hour (386 ppm) are significantly different. It is important to point out in Figure 3.4c the presence of a less probable source (~0.025) located to the north that is associated with slightly slower wind speeds, just less than 1 m/s. This may indicate a smaller source emission. Since it is associated with slower wind speeds, it could be the presence of sewer access cover MH4 emissions.

Both the morning and evening CH₄ CBPFs suggest an emission source located to the west-northwest direction, at which a classroom/office building is located. It appears that the main source of CO₂ emissions during rush hour is 100 S./North Campus Dr. and the main source of CH₄ at the same time is building emissions.

After spikes in concentration were observed in the early morning hours of the times

series from TL1 and FASB1, data collected from 4 am to 7 am were analyzed using wind roses and CBPF plots. Initially, this was attributed to campus maintenance operations initiating for the day and buildings starting their heating systems. The wind rose in Figure 3.5 shows the predominant wind direction coming from the northeast. The CO₂ CBPF (Figure 3.6a) suggests a probable source emission located just south-southeast of the tower. As mentioned previously, the University Union Building and parking lot are located in that direction. The CH₄ CBPF during this time indicates a probable source emission to the northeast of the tower (Figure 3.6b). A parking lot and engineering buildings with active classrooms and laboratories are located in this direction.

3.2 FASB1 Diagrams

FASB1 is located approximately 250 feet northwest from TL1. The day and night wind roses for the overnight data collection period at FASB1 (Figure 3.7) and the 99th percentile plots (Figure 3.8) suggest that the predominant wind direction is from the north, in contrast to TL1. During the day, the north-northwest direction exhibited its second highest frequency, while the north-northeast direction had the second highest frequency at night. The effect of the diurnal mountain wind system is less obvious at FASB1 than it was at TL1. Large buildings just west of the FASB1 site are most likely the reason for dampening the diurnal mountain wind system effect on wind direction. FASB1 day and night 99th percentile concentrations are less variant than those recorded at TL1. Data were recorded at FASB1 from 4:30pm on a Friday (10/18/2013) until about 6pm on the following Saturday (10/19/2013). There was most likely much less

activity on campus during the daytime hours (7am to 7pm) during this data collection period because it was a Saturday. The most probable source of daytime CO₂ and CH₄ emissions is just to the north-northeast directions and is associated with wind speeds between 0.5 to 1 m/s. This source is likely either the MH4 sewer access cover or the 100 S./North Campus Dr. The nighttime CO₂ CBPF points to 100 S./North Campus Dr. as the source emission and the nighttime CH₄ CBPF suggests that the building to the west-northwest of the tower is the source emission.

Only one rush hour time period occurred during data collection at FASB1, which was Friday night from 4pm to 6pm. The wind rose diagram suggests that the predominant wind is coming from the north (Figure 3.9). Interestingly, the highest probability values for CO₂ and CH₄ sources were not in the direction of the road (Figure 3.10). The location of the most probable high concentration CO₂ source was to the east and the CH₄ source location was to the southeast. East of FASB1 is a grassed lawn and a building approximately 300 feet away. The Union parking lot is located approximately 250 feet southeast of FASB1. Sewer access cover MH3 is also located adjacent to the parking lot at approximately 240 feet.

Similar early morning trends were observed at FASB1 as were at TL1 and a period of higher CO₂ and CH₄ concentrations was observed. The wind rose diagram (Figure 3.11) shows the predominant wind direction coming from the north and north-northeast direction. The CBPF for CO₂ (Figure 3.12) shows a relatively high probability (0.8) that 99th percentile concentrations of CO₂ are coming from the north-northwest direction, where 100 S./North Campus Dr. is located, and are associated with winds between 1 and 1.5 m/s. The probable source location of the 99th percentile CH₄ concentrations is

to the west-northwest of the tower (Figure 3.12). The 4am to 7am CBPFs for CO₂ and CH₄ are similar to the nighttime CBPFs in Figure 3.8. It is also noteworthy that the CBPFs in Figures 3.8 and 3.12 have almost identical concentrations at the 99th percentile threshold.

For three consecutive days from March 6th to March 8th, 2014, the tower and instruments were sited at FASB1 and operated between approximately 11am and 5pm. The predominant wind direction was from the north (Figure 3.13). According to the CBPFs, the probable location of the CO₂ and CH₄ sources is to the north of the tower (Figure 3.14), which is in the direction of 100 South/North Campus Dr and bus stop is located. Since data collection in March of 2014 was limited to between 11am and 5pm, no temporal subsets were explored.

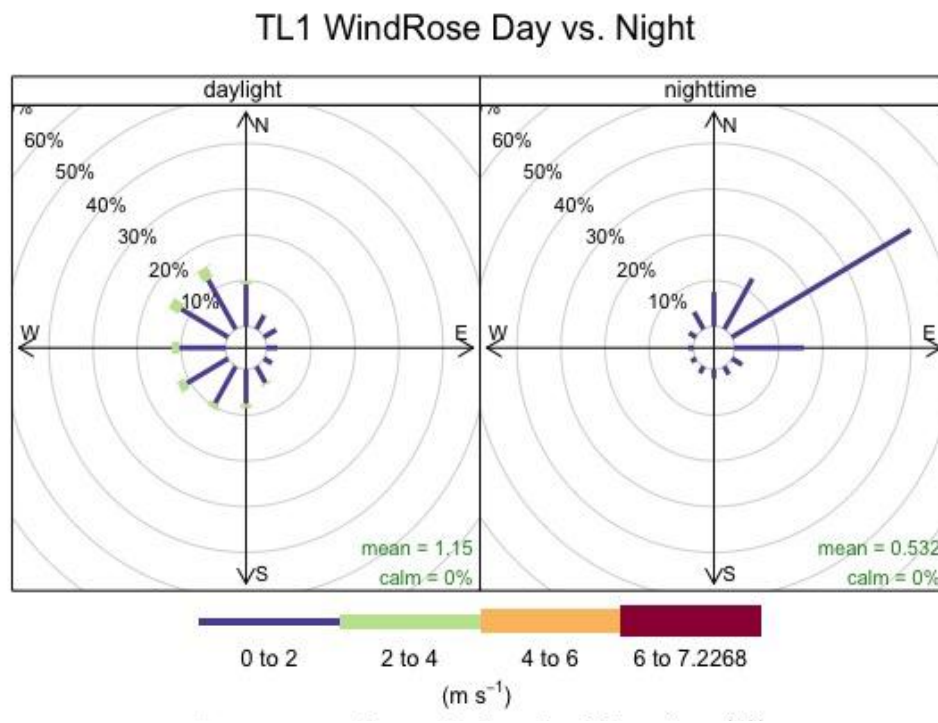


Figure 3.1 TL1 daytime and nighttime wind roses

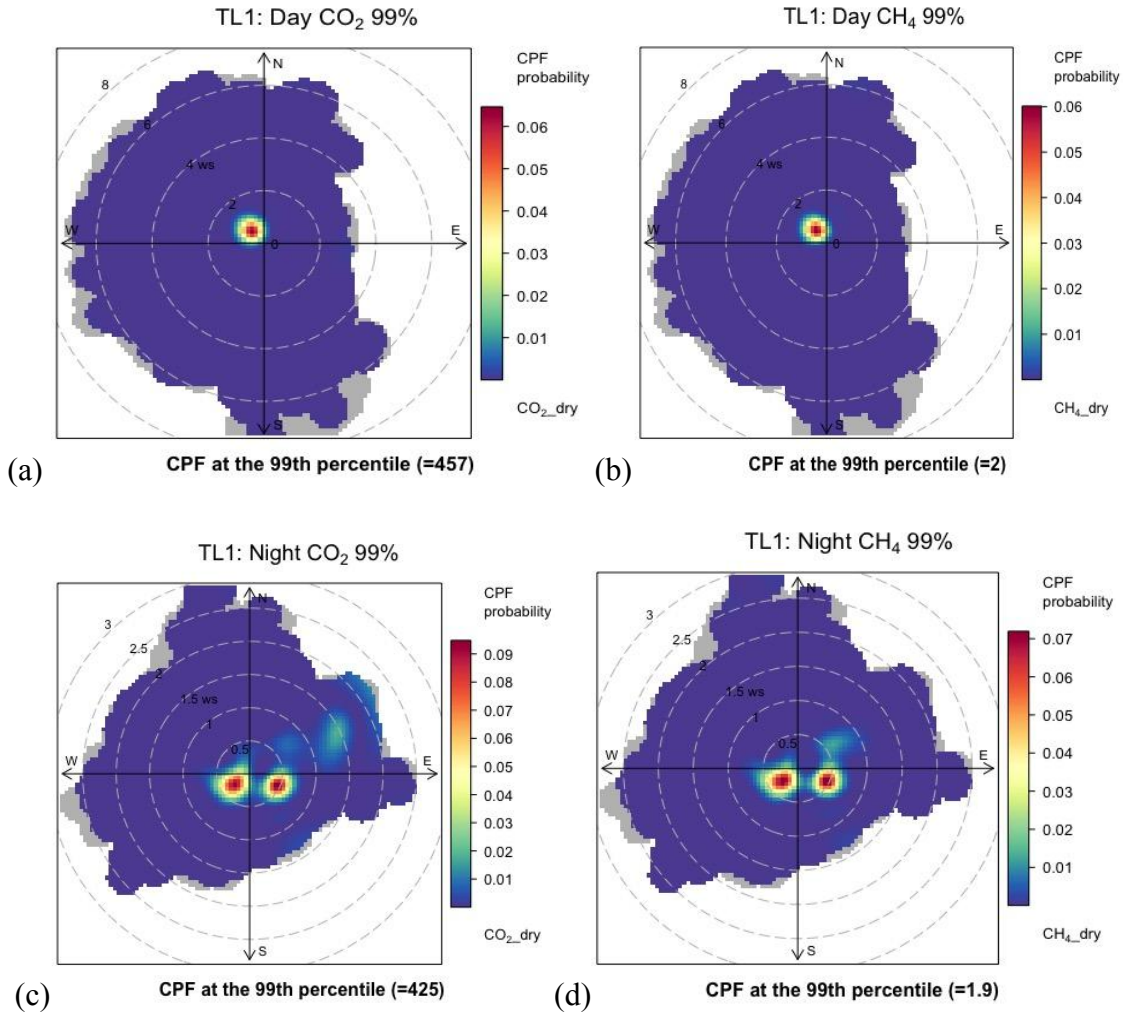
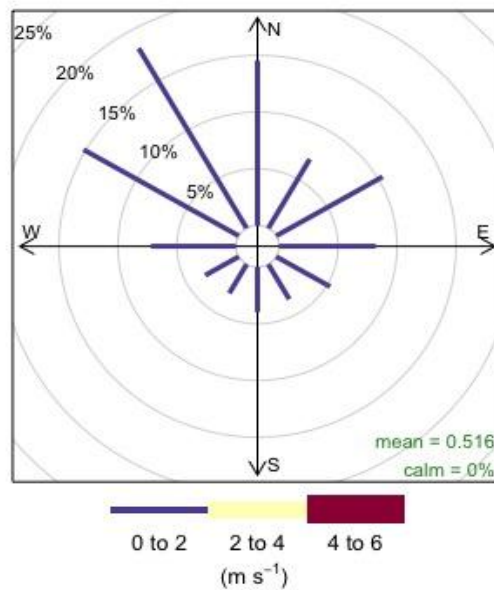


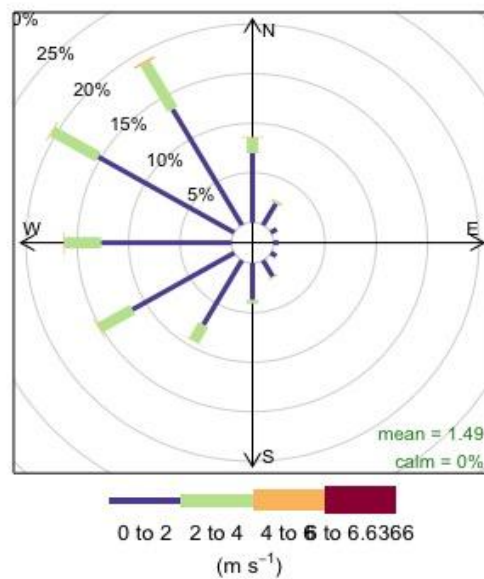
Figure 3.2 TL1 CBPF plots. (a) 99th perc. CO₂ during the day, (b) 99th perc. CH₄ during the day, (c) 99th perc. CO₂ during the night, (d) 99th perc. CH₄ during the night.

TL1: Wind Rose AM Rush Hour



(a)

TL1: Wind Rose PM Rush Hour



(b)

Figure 3.3 TL1 wind roses. (a) AM rush hour wind rose, (b) PM rush hour wind rose. AM rush hour was determined to be between 7am and 10am in the morning. PM rush hour was set as 4pm to 6pm in the evening.

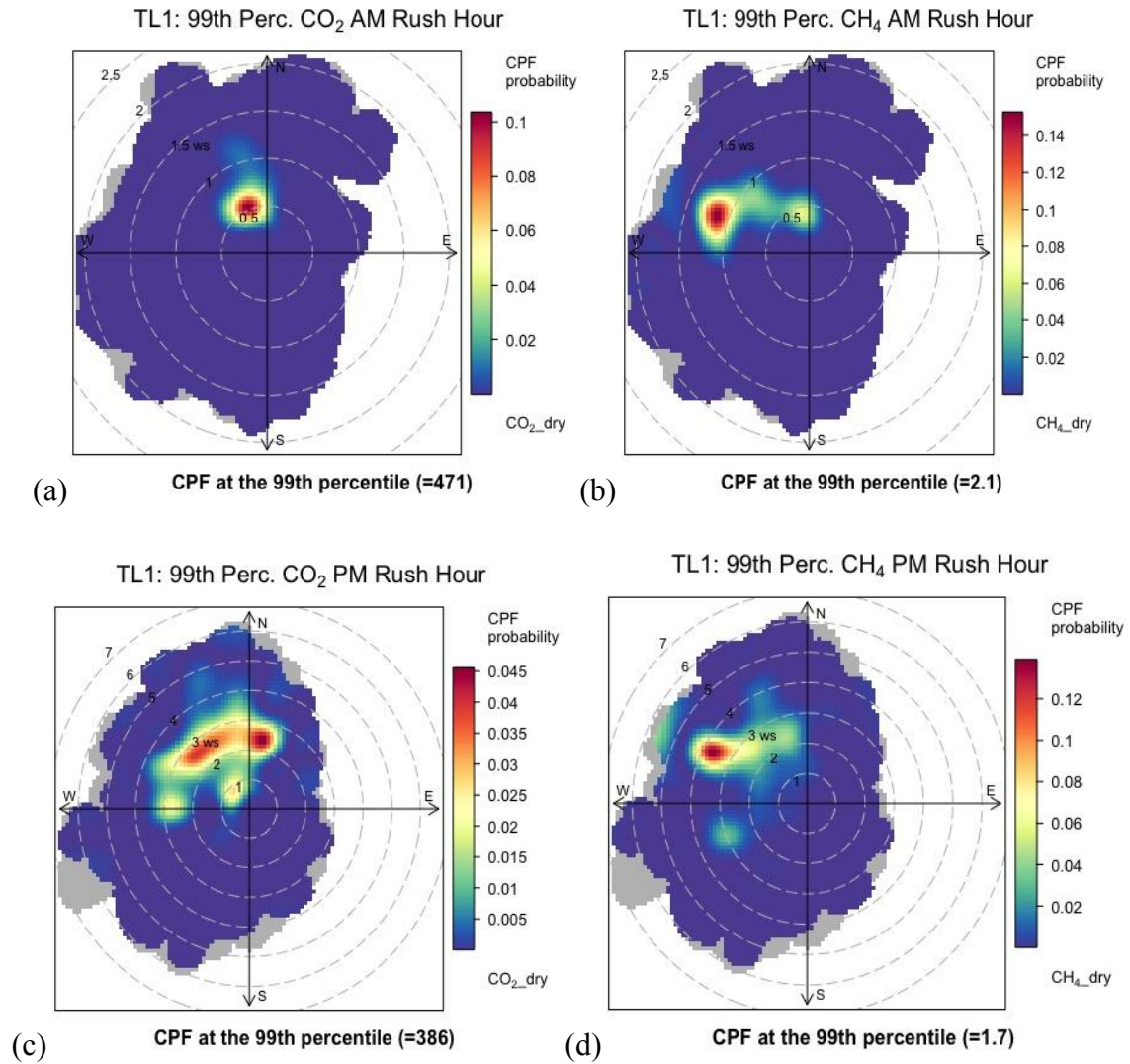


Figure 3.4 TL1 CBPF plots. (a) 99th perc. CO_2 during AM rush hour, (b) 99th perc. CH_4 during AM rush hour, (c) 99th perc. CO_2 during PM rush hour, (d) 99th perc. CH_4 during PM rush hour.

TL1: Wind Rose 4AM-7AM 10/18/13

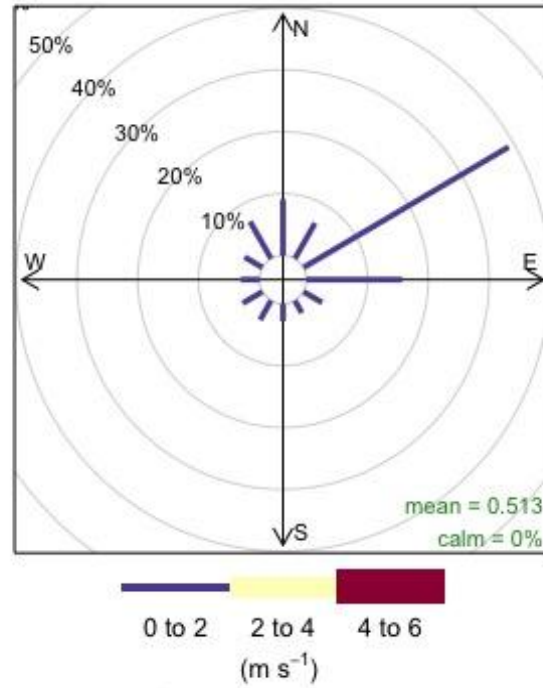


Figure 3.5 TL1 early morning wind rose

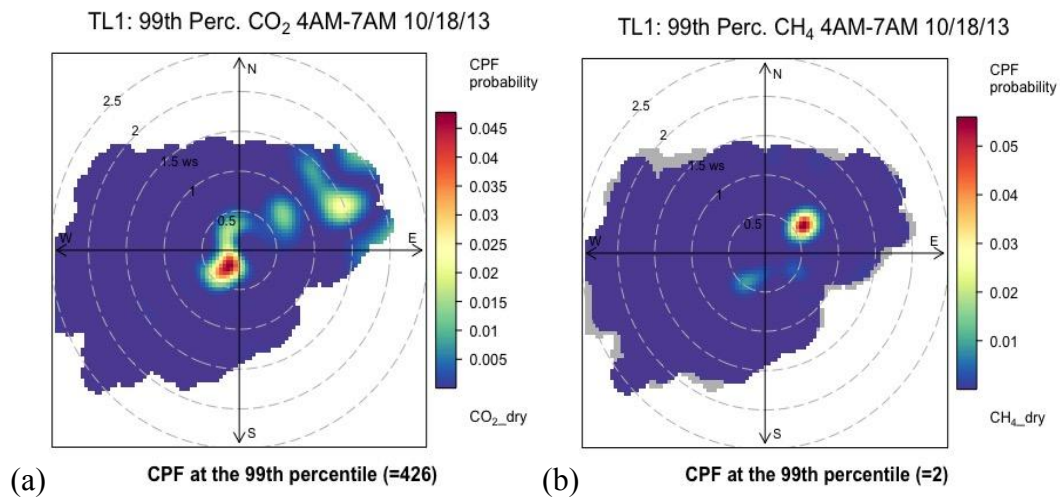


Figure 3.6 TL1 early morning (4am to 7am) CBPF plots. (a) 99th perc. CO₂, (b) 99th perc. CH₄.

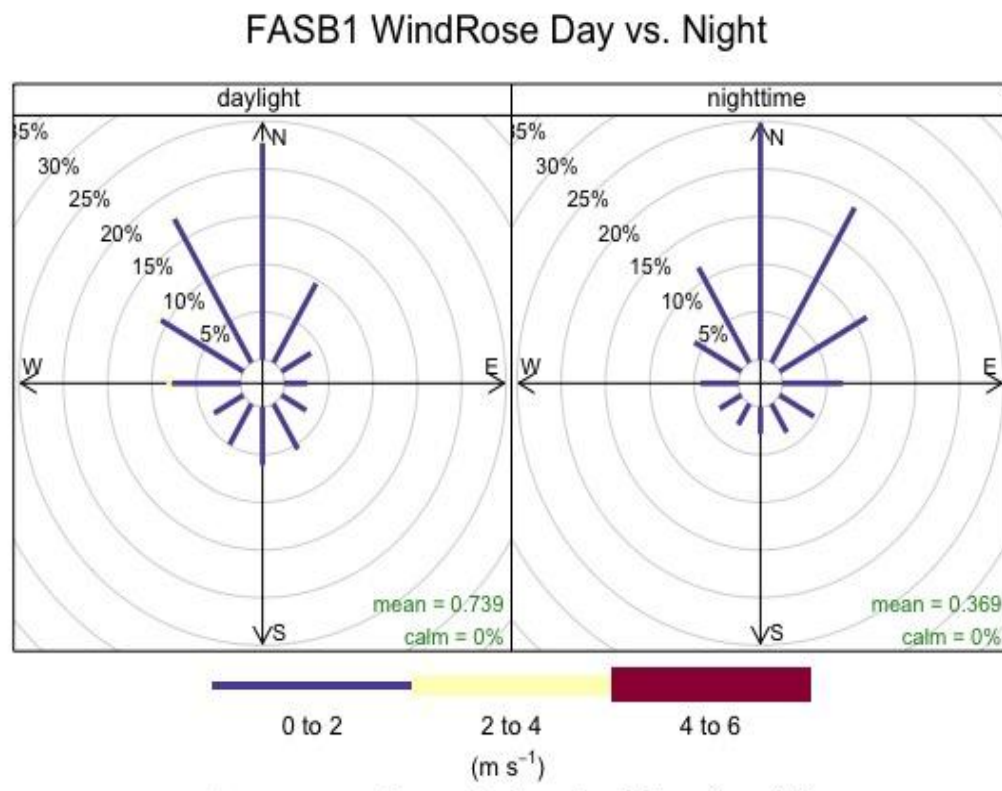


Figure 3.7 FASB1 daytime and nighttime wind roses

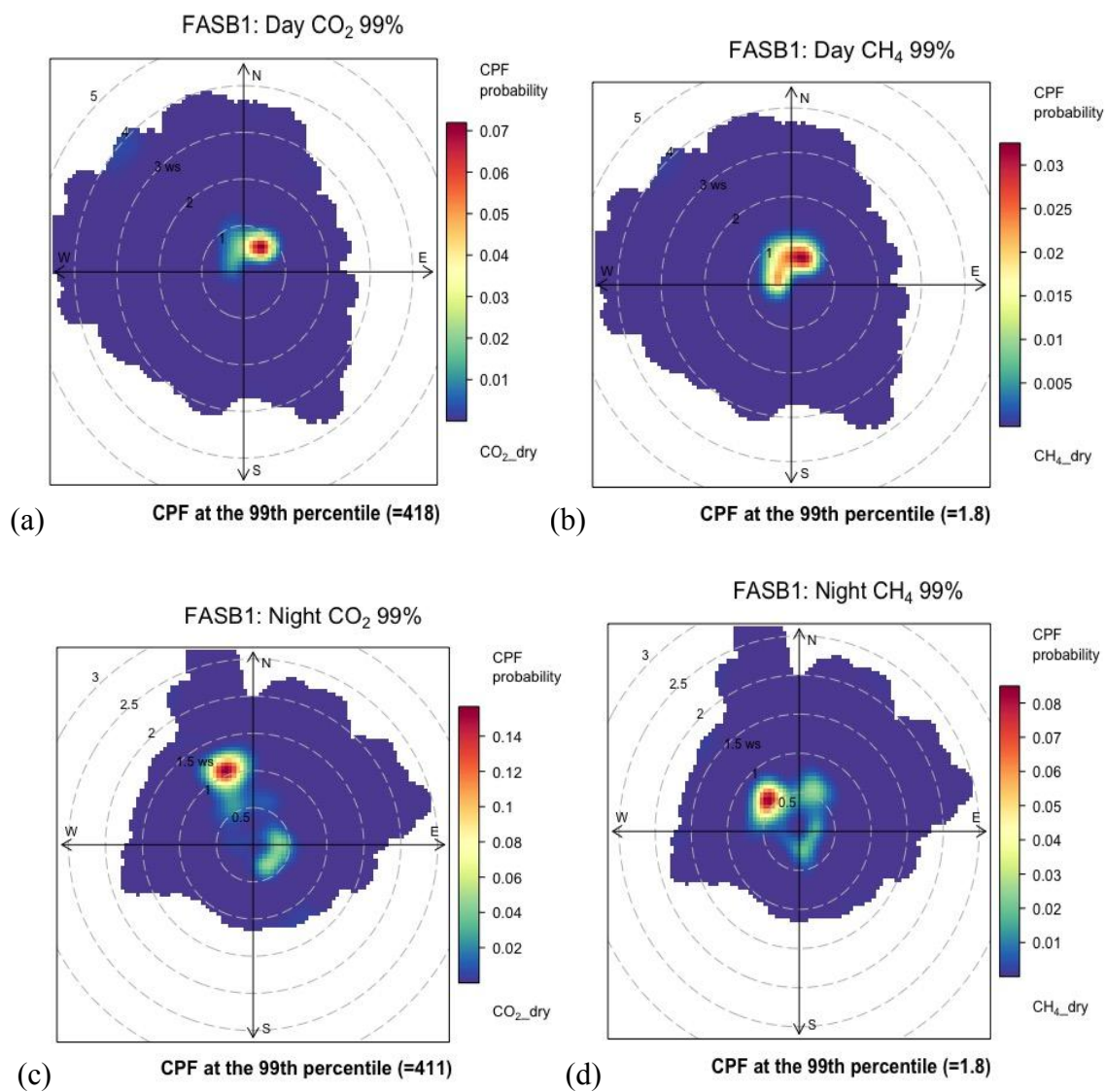


Figure 3.8 FASB1 CBPF plots. (a) 99th perc. CO₂ during the day, (b) 99th perc. CH₄ during the day, (c) 99th perc. CO₂ during the night, (d) 99th perc. CH₄ during the night.

FASB1: Wind Rose Rush Hour 10/18/13

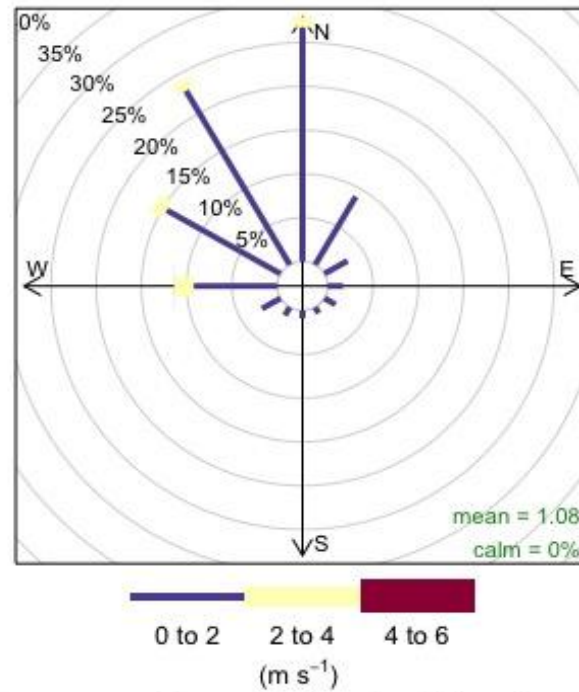


Figure 3.9 FASB1 rush hour wind rose

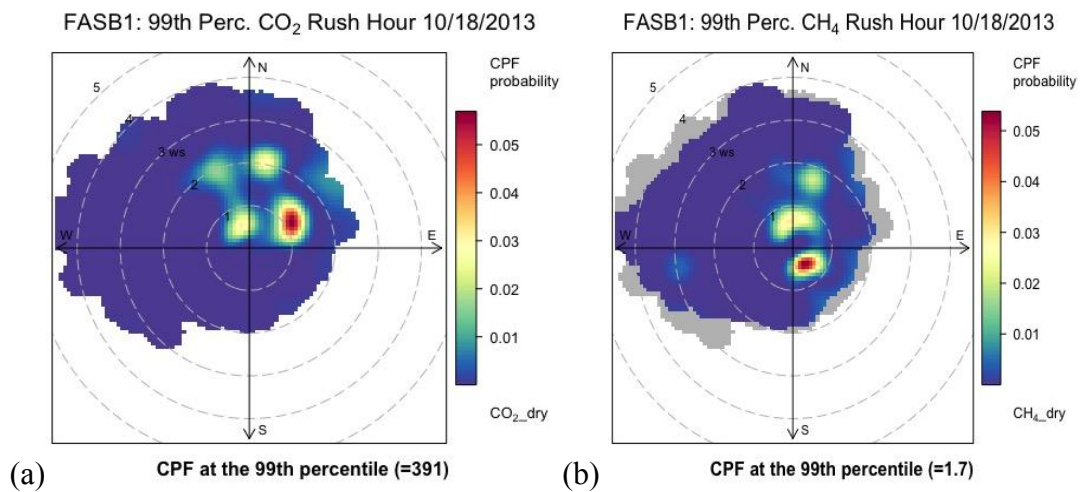


Figure 3.10 FASB1 evening rush hour CBPF plots. (a) 99th perc. CO₂, (b) 99th perc. CH₄.

FASB1: Wind Rose 4AM-7AM 10/19/13

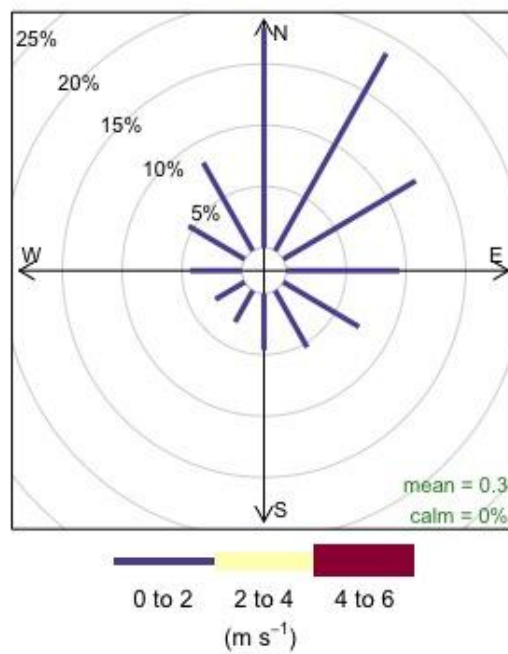
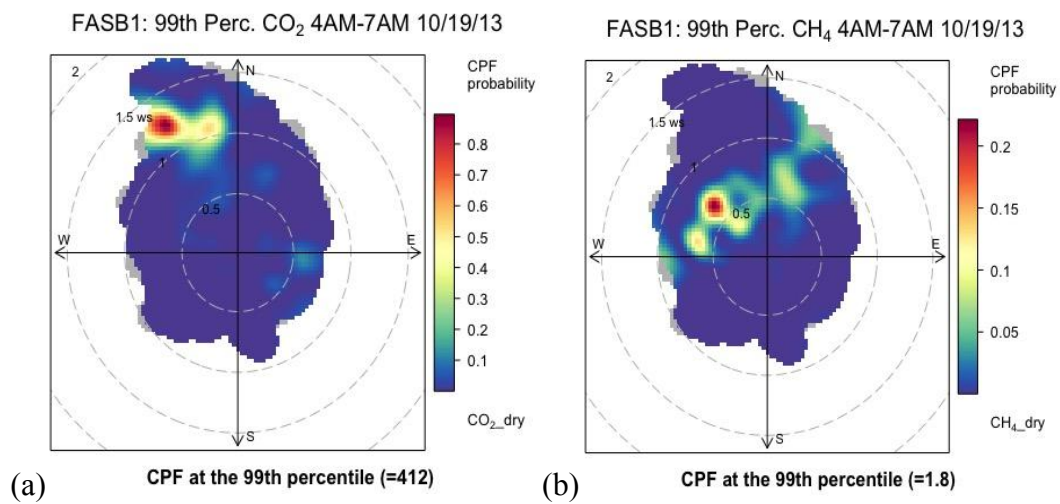


Figure 3.11 FASB1 early morning wind rose

Figure 3.12 TL1 early morning CBPF plots. (a) 99th perc. CO₂, (b) 99th perc. CH₄.

FASB1 March 2014 Wind Rose

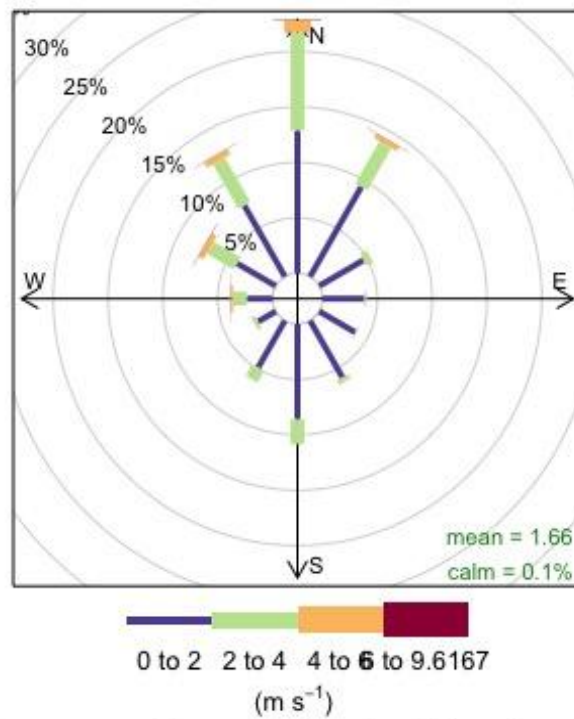


Figure 3.13 Wind rose diagram for data collected in March 2014 at FASB1

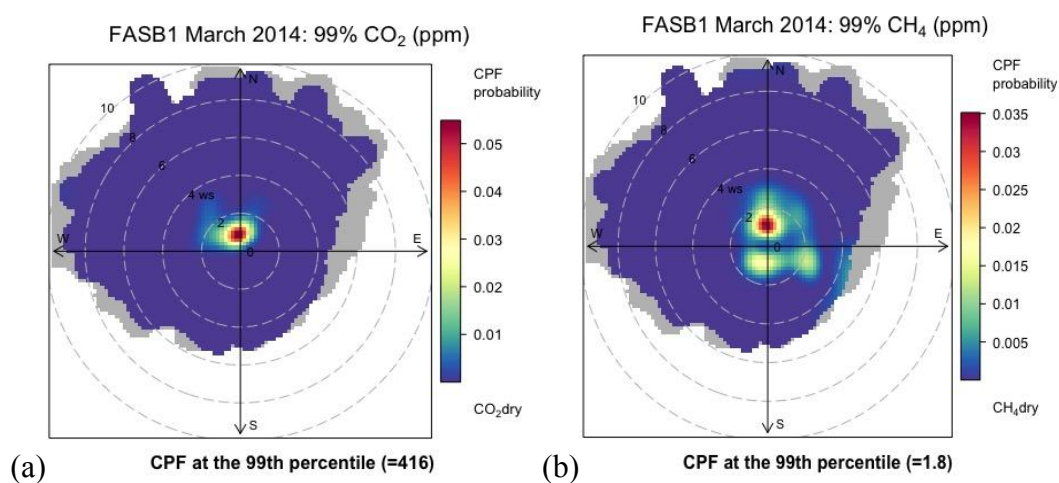


Figure 3.14 CBPFs for data collected at FASB1 during March 2014. (a) 99th perc. CO₂, (b) 99th perc. CH₄.

CHAPTER 4

CONCLUSIONS

This application suggests that different GHG emission sources may be identified using shorter duration datasets, on the order of 1-2 days in length. This work also demonstrates how CBPFs can be applied to shorter timespans within a dataset, such as morning and afternoon rush hour, to provide a “snapshot” analysis of temporal variations of probable emission source locations. Previous studies plotted CBPFs over different concentration ranges to pinpoint probable source locations (Uria-Tellaetxe and Carslaw, 2014; Krecl et al., 2016). These studies used longer periods of data collected at a lower frequency. Even though the timespan of data collection for this study on the University of Utah campus was relatively short, the data used to plot the CBPF diagrams was appropriate due to the high frequency of data collection leading to a large, relatively high resolution dataset. This said, however, a longer duration of samples is ideal to determine if the results presented here are anomalous (unique to anomalous conditions) or if they robustly reflect temporal trends in pollution source location. With a longer duration of samples, even more trends may be observed, such as weekend GHG concentrations versus weekday. Furthermore, this study confirmed that there were several significant sources of GHG; however, the CBPFs revealed that the sewer access covers were not the only significant sources. Roadways, parking lots, and buildings were also probable

emission sources. Thus, future studies should consider the location of the tower and consider periodically moving the tower location in an effort to discern emission sources.

The CBPF plots suggest that the pollutant source location varied throughout the day. The variation in emission source direction is most likely due to changes in on-road traffic, parking lot traffic, and subsurface sewer line activity. This study used the 99th percentile concentration for each dataset as opposed to a set concentration value to account for the range of values of each subset dataset. Furthermore, the use of a percentile threshold is typical for leak source identification studies (Lewicki et al., 2009; Yang et al., 2011). This allowed examination of only the highest concentrations in each dataset. The daytime and nighttime CO₂ and CH₄ CBPFs for both TL1 and FASB1 indicate that the 99th percentile concentration levels vary, especially for CO₂. The 99th percentile concentration levels are lower for the FASB1 dataset, most likely due to the majority of data collection taking place on a Saturday when anthropogenic GHG emissions are lower as a result of reduced campus activity. Furthermore, FASB1 CBPFs do not show probable source locations towards the direction of the Union building or Union parking lots (south and southeast directions), corroborating the interpretation that campus activity impacts CO₂ and CH₄ emission source location estimates. Analysis of longer sets of data will be helpful to test this hypothesis.

The CBPF plots revealed that, overall, 99th percentile concentrations are associated with lower wind speeds (<1.5 m/s). We suggest that higher GHG concentrations are diluted by high wind speeds and associated turbulent patterns before tangible signals can reach the tower. Furthermore, nearby buildings, trees, and the hilly terrain of the study site contributed to dispersion of CO₂ and CH₄ signals coming from an emission source.

The location of the tower with respect to the emission source location coupled with dominant wind direction is crucial to any such receptor study. Study sites located at or near the mouth of a canyon can be affected by diurnal mountain wind systems; such periodic effects should be highlighted, if not “corrected” or filtered. Examining snapshots of data during specific time periods throughout the day to determine (assess) multiple likely emission sources, as done in this study, also helps determine where towers should be located for subsequent experimental studies. Specifically, future studies at the University of Utah would benefit by locating a tower on the other side of 100 S./North Campus Dr. Plotting CBPFs using data collected from another direction with respect to the suspected source would probably confirm whether or not 100 S./North Campus Dr. is a main GHG emission source.

For future studies using a Picarro[®] CRDS or similar instrument in conjunction with a 3-D anemometer, it is suggested that more than one tower be used to verify probable emission source locations. Furthermore, with multiple towers, a conditional probability map could be created by triangulating CPFs from each tower, similar to a method employed by Sung et al. (2014).

CHAPTER 5

FUTURE DIRECTIONS

In effort to continue the work described herein, a GHG monitoring tower was set up on the University of Utah Ray Olpin Student Union building in June of 2016 as a case study looking into the effectiveness of using the CBPF method to determine the location of GHG emission sources. Specifically, at least one restaurant in the Union building appears to emit both CO₂ and CH₄ at significantly elevated rates. Our hypothesis is that cook stoves in this specific restaurant emit CO₂ as a byproduct of combustion of natural gas (CH₄), but because the stoves do not burn all CH₄ completely, some CH₄ is also emitted through the restaurant flues. Once the Union building roof experiments are completed, the monitoring equipment will be sent to a geologic carbon capture & sequestration (CCS) demonstration site to monitor atmospheric CO₂ and CH₄. Appendix B describes the work that is currently underway on the Union roof in more detail as well as some of the preliminary findings.

For the Union building roof research study, an LI-7500A Open Path CO₂/H₂O Analyzer, LI-7700 Open Path CH₄ Analyzer, Licor Smartflux[®] onsite processing unit and data-logger, and a Gill HS-50 three-dimensional (3D) anemometer are being used. The instruments were attached to an adjustable tripod tower. In addition to CO₂, CH₄, H₂O, and wind data, the system also collects temperature and pressure from both the

Smartflux[®] unit and the LI-7700 unit. The Licor tower was set up on the University of Utah Ray Olpin Student Union building in June of 2016. Instruments have been continuously collecting CO₂, CH₄, H₂O, and wind data since that time. The Union building roof was used as the study site because it is expected that the exhaust outlets from the restaurant kitchens would be characterized by a higher CO₂ and CH₄ signal during hours of operation (approximately 9am to 5pm).

APPENDIX A

R SCRIPTS FOR DATA ANALYSIS ON UNIVERSITY OF UTAH

CAMPUS (TL1 AND FASB)

A.1 TL1 Script

This script takes the data from the Picarro © CRDS raw data files for site TL1, formats the data, accounts for lag time, performs statistics, and creates the CBPF plots.

```
setwd("~/Dropbox/Research/MinckData/TLI_Oct17_18_LagFixed")
#setwd("C:/Users/Meg/Dropbox/Research/MinckData/TLI_Oct17_18_LagFixed")
#On Server: setwd("/Volumes/mwalsh/Documents/MinckData/TLI_Oct17_18_LagFixed")
list.files(path = ".", pattern = NULL, all.files = FALSE, full.names = FALSE,
recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALSE)
TL1<-read.table("~/Dropbox/Research/MinckData/Raw/TL1_HP20131017.txt",
header=TRUE, sep = ',')

names(TL1)
class(TL1)
str(TL1)
class(TL1$DATE) #Returns type of class of that column
TL1$DATE[1] #returns first date
tail(TL1, 10) #prints last 10 rows
head(TL1, 20) #prints first 20 rows
# === !! Note that the first 6445 rows don't have anemometer data!!!!
TL1 <-TL1[-c(1:6445), ] #This removes the first 6445 rows of dataset
# ===== Note that there is a strange peak of CO2 at 9:02, start data set at
9:04
TL1<-subset(TL1, DT>"2013-10-17 09:04:00.000 MDT")

## ===== Time date stuff =====
## Note that the Picarro time is in GMT.
# ===== create date time column
```

```

library(lubridate)
op <- options(digits.secs = 3) ##Need to set this
TL1$DT<-as.POSIXct(strftime(paste(TL1$DATE, TL1$TIME), format="%Y/%m/%d
%H:%M:%OS3"), tz="GMT")
TL1$DT<-with_tz(TL1$DT, tz="America/Denver")
hour(TL1$DT[1]) # returns just hour of date-timestamp
## Also note that we need to adjust for a lag time of 27.7 seconds.
library(lubridate)
start<-TL1$DT[1]
end<-TL1$DT[length(TL1$DT)]
span<-interval(start,end)
as.duration(span)
as.period(span, unit="hours")

##count how many times CH4_dry or CO2_dry = 0
TL1$CH4_dry==0
sum(TL1$CH4_dry == 0) #determines how many times 0 value occurs
which(TL1$CH4_dry==0) #Returns position of 0 values
TL1$CO2_dry==0
sum(TL1$CO2_dry == 0) #determines how many times 0 value occurs
which(TL1$CO2_dry==0) #Returns position of 0 values

## WIND Calculations
TL1$WindSpeed=sqrt(TL1$ANEMOMETER_UX^2 + TL1$ANEMOMETER_UY^2 +
TL1$ANEMOMETER_UZ^2) #units=m/s
TL1$N = TL1$ANEMOMETER_UX;
TL1$E = TL1$ANEMOMETER_UY;
TL1$D = (180/pi)*atan2(TL1$E,TL1$N); #converted to degrees
##Need to get rid of negative values
##### instead of I statements, use this!
idx1 <- TL1$D < 0
idx2 <- TL1$D > 0
idx3 <- TL1$D == 0
TL1$D1[idx1] <- TL1$D[idx1] * (-1)
TL1$D1[idx2] <- (TL1$D[idx2] * (-1)) + 360
TL1$D1[idx3] <- 0

OFFNDEG<-135 #offset angle for anemometer at TL1
idx4 <- (360-TL1$D1)<=OFFNDEG
idx5 <- (360-TL1$D1)>OFFNDEG
TL1$DEG[idx4] <- (TL1$D1[idx4]-360+OFFNDEG)
TL1$DEG[idx5] <- (TL1$D1[idx5]+OFFNDEG)
any(is.na(TL1$DEG)) # make sure there are no NA's
write.table(TL1,
"~/Dropbox/Research/MinckData/TLI_Oct17_18_LagFixedTL1_windDir.txt", sep=",")
# write table because sometimes Rstudio crashes

# ===== Graphics =====#
hist(TL1$DEG, 72, main="Wind Direction, TL1, Oct. 17-18, 2013",
col="grey",xlab="Direction (Degrees)")
hist(TL1$CO2_dry, breaks=50, main="CO2 dry, TL1, Oct. 17-18, 2013",
col="grey",xlab="CO2 dry (ppmv)")
abline(v=quantile(TL1$CO2_dry, c(.99)), col="red", lwd=3) #adds 99th
percentile
hist(TL1$CH4_dry, breaks=60, main="CH4 dry, TL1, Oct. 17-18, 2013",

```

```

col="grey",xlab="CH4 dry (ppmv)")
abline(v=quantile(TL1$CH4_dry, c(.99)), col="red", lwd=3) #adds 99th
percentile

# Circular plot#
###try subestting data
sample<-TL1[ sample(TL1$DEG), round(0.0001*length(TL1$DEG)), ]
  round(0.0001 * nrow(mydf[mydf$gender == "F",])) #.01% of wind deg data
sample_deg<-sample(TL1$DEG,1000,replace = FALSE)

WindDir<-circular(TL1$DEG)
#rose.diag(WindDir, type="angles", units = 'degrees', zero = pi/2) ## Rotates
where 0deg is

## Statistics
summary(TL1$CO2_dry)
summary(TL1$CH4_dry)
summary(TL1$WindSpeed)
quantile(TL1$CO2_dry, c(.99)) # 99th percentile
quantile(TL1$CH4_dry, c(.99)) # 99th percentile

## Standard Deviation
sd(TL1$CO2_dry)
sd(TL1$CH4_dry)

## Plotting
# Note: dev.off() clears plotting window
boxplot(TL1$CO2_dry, cex=16, ylab="CO2 Dry (ppmv)", main="6/1/2015 CO2dry")
boxplot(TL1$CH4_dry, cex=16, ylab="CH4 Dry (ppmv) 6/1/2015")
plot(TL1$CO2_dry, type='h')
## HISTOGRAMS
hist(TL1$CO2_dry)
hist(TL1$CO2_dry, breaks=75, cex=16, xlab="CO2 Dry (ppmv)", main="6/1/2015
CO2dry")
boxplot(TL1$CH4_dry, cex.axis=1.4, ylab="CH4 Dry (ppmv) 6/1/2015")

#===== Circular Statistics =====
library(circular)
Tl1circ<-circular(TL1$DEG, units = 'degrees', zero = pi/2, rotation="counter")
mean(Tl1circ)
MeanResultLength<-rho.circular(Tl1circ)
CircSD_TLI<-sd.circular(Tl1circ)
V<-(1-MeanResultLength) #Variance

# ===== Linear-Circular Association
TL1_Radians<-TL1$DEG*2*pi/360
R2xtIndTestRand(TL1$CO2_dry,TL1_Radians,9999) #Johnson-Wehrly-Mardia
Corelation COefficient
# [1] 0.002859866 0.000100000
uniscres<-OrderScores(TL1$CO2_dry,TL1_Radians) #All CO2
MardiaRankIndTestRand(uniscres, 9999)

```

```

uniscores<-OrderScores(TL1$CH4_dry,TL1_Radians) #All CH2
MardiaRankIndTestRand(uniscores, 9999)

#===Subset daa to 99th percentile CO2
CO2dry99<-quantile(TL1$CO2_dry, c(.99))
TL199CO2 <- TL1[ which(TL1$CO2_dry >= CO2dry99), ]
TL199circ<-circular(TL199CO2$DEG, units = 'degrees', zero = pi/2,
rotation="counter")
mean(TL199circ)
MeanResultLength99<-rho.circular(TL199circ)
CircSD_TL199<-sd.circular(TL199circ)
V_TL199<-(1-MeanResultLength99) #Variance

# ===== Linear-Circular Association
TL199_Radians<-TL199CO2$DEG*2*pi/360
R2xtIndTestRand(TL199CO2$CO2_dry,TL199_Radians,9999) #Johnson-Wehrly-Mardia
Corelation
uniscores<-OrderScores(TL199CO2$CO2_dry,TL199_Radians)
MardiaRankIndTestRand(uniscores, 9999)

# ===== Rayleigh Test of significance of mean direction (pg 81 Pewsey)
rtest<-rayleigh.test(TL1circ)
rtest

#===Subset data to 99th percentile CH4
CH4dry99<-quantile(TL1$CH4_dry, c(.99))
TL199CH4 <- TL1[ which(TL1$CH4_dry >= CH4dry99), ]
TL199CH4circ<-circular(TL199CH4$DEG, units = 'degrees', zero = pi/2,
rotation="counter")
mean(TL199CH4circ)
MeanResultLength99CH4<-rho.circular(TL199CH4circ)
CircSD_TL199CH4<-sd.circular(TL199CH4circ)
V_TL199CH4<-(1-MeanResultLength99CH4) #Variance
# ===== Linear-Circular Association
TL199CH4_Radians<-TL199CH4$DEG*2*pi/360
uniscores<-OrderScores(TL199CH4$CH4_dry,TL199CH4_Radians)
MardiaRankIndTestRand(uniscores, 9999)
# ===== Rayleigh Test of significance of mean direction (pg 81 Pewsey)
rtestCO2<-rayleigh.test(TL199circ) #CO2
rtestCH4<-rayleigh.test(TL199CH4circ) #CH4

## ===== Timeseries
library(xts)
zooTL1CO2<-zoo(TL1$CO2_dry,TL1$DT) #use zoo becuase they are not constant time
intervals
plot(zooTL1CO2, main = "CO2dry TL1, Oct. 17-18, 2013", ylab="CO2dry (ppmv)",
xlab=NULL) # this needs some formatting
abline(h=CO2dry99, col="red", lwd=3) #adds 99th percentile

plot(zoo, xaxt = "n",main = "CO2dry Tl1, Oct. 17-18, 2013", ylab="CO2dry
(ppmv)", xlab="")
axis(1, Tl1$DT, format(Tl1$DT, "%d %T"), las=2, cex.axis = .7)

zooTL1CH4<-zoo(TL1$CH4_dry,TL1$DT) #use zoo becuase they are not constant time
intervals

```

```

plot(zooTL1CH4, main = "CH4dry TL1, Oct. 17-18, 2013", ylab="CH4dry (ppmv)",
xlab=NULL) # this needs some formatting
abline(h=CH4dry99, col="red", lwd=3)

# ==== try combining both into one plot
TL1_TS<-cbind(TL1$CO2_dry, TL1$CH4_dry, TL1$WindSpeed)
TL1_TSzoo<-zoo(TL1_TS, TL1$DT)
summary(TL1_TSzoo)
plot(TL1_TSzoo, plot.type="multiple", main="TL1 Oct. 17-18, 2013",xlab="",
ylab=list("CO2 dry(ppmv)", "CH4 dry(ppmv)", "Wind Speed (m/s)"))
abline(h=CO2dry99)

# ===== SImple statistics/Modeling =====
#remove some of the unused variables, or find a way to designate certain
paramters
cor(TL199CO2)
cov(TL199CO2)
cor.test(TL199CO2$CO2_dry, TL199CO2$DEG, method="spearman")
cor.test(TL199CO2$CO2_dry, TL199CO2$WindSpeed, method="spearman")
plot(TL199CO2)
colnames(TL199CH4) #Columns names
pairs(TL199CH4[,c(4,6,7,13,18)], main="99th perc. CH4dry TL1") # sepcific
columns

#=== Training testing datasets
set.seed(10000)
train_idx <- sample(1:nrow(TL1),1000,replace=FALSE)
train <- TL1[train_idx,] # select all these rows
test <- TL1[-train_idx,] # select all but these rows

#=== Building CARTs
library(rpart)
library(tree)
library(randomForest)
TL1CO2.rpart<-rpart(CO2_dry~DEG+WindSpeed+TIME+CH4_dry, data=train)
plot(TL1CO2.rpart)
text(TL1CO2.rpart, cex=0.75)
plotcp(TL1CO2.rpart)
plot(predict(TL1CO2.rpart), residuals(TL1CO2.rpart), xlab="Fitted values",
ylab="Residuals")
abline(h=0)
PredTL1CO2=predict(TL1CO2.rpart, test)
plot(PredTL1CO2)
# Try to get a table of observed vs predicted
table(Original = train$CO2_dry,Predicted = predict(TL1CO2.rpart))

#===== KRUSKAL-WALLIS TEST =====#
kruskal.test(CO2_dry ~ DEG, data = TL199CO2)
#data: CO2_dry by DEG
#Kruskal-Wallis chi-squared = 36385, df = 35257, p-value = 1.301e-05

```

```

# ===== Prob ROse Plots and Polar Histograms =====#
#===Breaking up degrees into bins
TL1$bin<-trunc(0.2*TL1$DEG)+1
TL1$C0299<-TL1$C02_dry >= C02dry99 #If C02 is >= 99th percentile, returns a
true
TL1$CH499<-TL1$CH4_dry >= CH4dry99 #If C02 is >= 99th percentile, returns a
true
#===== Create new dataset with 72 rows for count data
TL1Count<-data.frame("bin"=numeric(72), "TotalCount"=numeric(72),
"Count99"=numeric(72),"Avg99Conc"=numeric(72))
TL1Count$bin=seq(1,72)
TL1Count$degrees=seq(0,355,5)
for(i in 1:length(TL1Count$bin)){ ##this counts total readings in each bin
  TL1Count$TotalCount[i]=sum(TL1$bin==i)
}
rm(i)
for(i in 1:length(TL1Count$bin)){ ## this counts total 99th perc reading in
each bin
  TL1Count$CountC0299[i]=sum(TL1$bin==i & TL1$C0299==TRUE)
}
rm(i)
TL1Count$ProbC02=TL1Count$CountC0299/TL1Count$TotalCount

for(i in 1:length(TL1Count$bin)){ ## this counts total 99th perc CH4
reading in each bin
  TL1Count$CountCH499[i]=sum(TL1$bin==i & TL1$CH499==TRUE)
}
rm(i)
TL1Count$ProbCH4=TL1Count$CountCH499/TL1Count$TotalCount

plot(TL1Count$degrees, TL1Count$ProbC02, pch=18, col='red', xlab='Bin Number',
ylab='99th Perentile C02 Probability', font.lab=2)
## == Polar Probability ROse Plot:
oldpar<-polar.plot(TL1Count$Prob,TL1Count$degree,main="Test: TL1, C02dry Prob
Rose", radial.lim=c(0,.05),start=90,clockwise=TRUE,lwd=4,line.col=4)

## =====Pollution COncentration Roses
CountMeanC02<-tapply(TL1$C02_dry, TL1$bin, mean) # this averages the C02
concentration by bin
TL1Count$AvgConcC02<-as.numeric(CountMean)
plot(TL1Count$degrees, TL1Count$AvgConcC02, pch=18, col='red', xlab='Bin
Number', ylab='Avg C02 Pollution', font.lab=2)

Count99MeanC02<-by(TL1$C02_dry, list(TL1$bin, TL1$C0299), mean) # this
averages the C02 concentration by bin & 99%=TRUE))
for(i in 1:length(TL1Count$bin)){
  TL1Count$Avg99Conc[i]=Count99MeanC02[i+72]
}
rm(i)
plot(TL1Count$degrees, TL1Count$Avg99Conc, pch=18, col='red', xlab='Bin
Number', ylab='Avg 99th Perentile C02 Pollution', font.lab=2)

# == Wind Frequency
TotWind<-sum(TL1Count$TotalCount)
TL1Count$WindFreq<-TL1Count$TotalCount/TotWind

```



```

TL1Count$C0299WindFreq<-TL1Count$ProbC02/TL1Count$WindFreq
TL1Count$CH499WindFreq<-TL1Count$ProbCH4/TL1Count$WindFreq
oldpar<-polar.plot(TL1Count$C0299WindFreq,TL1Count$degree,main="TL1 C02 Prob
Rose/AvgWindFreq",
radial.lim=c(0,2.5),start=90,clockwise=TRUE,lwd=4,line.col=3)

### ==== CONDITIONAL PROBABILITY -> P(99th|ni) , Jun 2016 ===== #####
### This calculates the probability that there is a 99th percentile reading in
### a given wind direction bin using conditional probability trees
TL1Count$Cond99C02=TL1Count$ProbC02*TL1Count$WindFreq
## the prob that there is a 99th % reading AND it is in a given bin
TL1Count$Cond99CH4=TL1Count$ProbCH4*TL1Count$WindFreq

# === OPENAIR
pollutionRose(TL1, pollutant ="C02_dry", ws="WindSpeed", wd="DEG",
breaks=c(300,350,400,450,500))
pollutionRose(TL199C02, pollutant ="C02_dry", ws="WindSpeed", wd="DEG",
breaks=c(450,460, 470,480,490,500))

#=== Conventional Pollution Plots
library(plotrix)
polar.plot(TL1Count$AvgConc,TL1Count$degree,main="TL1, C02dry Pollution Conc
Rose", radial.lim=c(395,405),start=90,clockwise=TRUE,lwd=4,line.col=4)
polar.plot(TL1Count$Avg99Conc,TL1Count$degree,main="TL1, 99th % C02dry
Pollution Conc Rose",
radial.lim=c(450,470),start=90,clockwise=TRUE,lwd=4,line.col=4)

## === Wind Histograms with GGplot2
library(ggplot2)
ggplot(data=TL1, aes(TL1$DEG)) +geom_histogram(binwidth = 5,fill=I("blue")) +
scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))
+
coord_polar(theta="x", start=0, direction =1)+
ggtitle("TL1 Oct. 17-18, 2013 Wind Direction")+
theme(plot.title = element_text(size=14, face="bold"))+
labs(x="", y="") + theme_bw() +
theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="plain", size=16),
panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))
# == wind histogram of just 99th percentile data for C02
ggplot(data=TL199C02, aes(TL199C02$DEG)) +geom_histogram(binwidth =
5,fill=I("palegreen4")) +
scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))
+
coord_polar(theta="x", start=0, direction =1)+
ggtitle("TL1 Oct. 17-18, 2013 99th percentile C02 Wind Direction")+
theme(plot.title = element_text(size=14, face="bold"))+
labs(x="", y="") + theme_bw() +
theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="bold"), axis.text.x=element_text(size=14,
face="bold"), axis.title.y=element_text(face="plain", size=16),
panel.grid.minor = element_line(colour="grey"),

```

```

panel.grid.major=element_line(colour = "grey"))
# == wind histogram of just 99th percentile data for CH4
ggplot(data=TL199CH4, aes(TL199CH4$DEG)) +geom_histogram(binwidth =
5,fill=I("yellowgreen")) +

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))
+
  coord_polar(theta="x", start=0, direction =1)+
  ggtitle("TL1 Oct. 17-18, 2013 99th percentile CH4 Wind Direction")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="bold"), axis.text.x=element_text(size=14,
  face="bold"), axis.title.y=element_text(face="bold", size=16),
  panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

# ==== ProbRose with ggplot2
TL1Count$deg_move<-TL1Count$degree+2.5 # had to move degrees by +2.5 to get
this to work
ggplot(data=TL1Count, aes(x=TL1Count$deg_move, y=TL1Count$ProbCO2))+
  geom_bar(stat="identity", fill="red")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("TL1 99perc CO2 ProbRose")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
  face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))
ggplot(data=TL1Count, aes(x=TL1Count$deg_move, y=TL1Count$ProbCH4))+
  geom_bar(stat="identity", fill="pink")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("TL1 99perc CH4 ProbRose")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
  face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

## ==== Conditional Prob Plots
ggplot(data=TL1Count, aes(x=TL1Count$deg_move, y=TL1Count$Cond99CO2))+
  geom_bar(stat="identity", fill="pink")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("TL1 Conditional Prob CO2")+
  theme(plot.title = element_text(size=14, face="bold"))+

```

```

    labs(x="", y="") + theme_bw() +
    theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))
ggplot(data=TL1Count, aes(x=TL1Count$deg_move, y=TL1Count$Cond99CH4))+
  geom_bar(stat="identity", fill="orange")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("TL1 Conditional Prob CH4")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

# === ggplot prob rose/wind freq
ggplot(data=TL1Count, aes(x=TL1Count$deg_move, y=TL1Count$C0299WindFreq))+
  geom_bar(stat="identity", fill="darkturquoise")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("TL1 99perc C02 ProbRose/WindFreq")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))
ggplot(data=TL1Count, aes(x=TL1Count$deg_move, y=TL1Count$CH499WindFreq))+
  geom_bar(stat="identity", fill="purple")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("TL1 99perc CH4 ProbRose/WindFreq")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

### ===== PEAK TIME PR
===== #####
# =====Subset dataframe based on time ===== #

```

```

PeakStart<-as.POSIXct(strptime("2013-10-18 04:00:00", "%Y-%m-%d %H:%M:%S"))
PeakEnd<-as.POSIXct(strptime("2013-10-18 07:00:00", "%Y-%m-%d %H:%M:%S"))
TL1_Peak<-subset(TL1, DT>PeakStart & DT<PeakEnd)
#==== Create new dataset with 72 rows for count data
TL1CountPeak<-data.frame("bin"=numeric(72), "TotalCount"=numeric(72),
"CountC0299"=numeric(72), "ProbC02"=numeric(72),"AvgConcC02"=numeric(72))
TL1CountPeak$bin=seq(1,72)
for(i in 1:length(TL1CountPeak$bin)){ ##this counts total readings in each
bin
  TL1CountPeak$TotalCount[i]=sum(TL1_Peak$bin==i)
}
rm(i)
for(i in 1:length(TL1CountPeak$bin)){ ## this counts total 99th perc C02
reading in each bin
  TL1CountPeak$CountC0299[i]=sum(TL1_Peak$bin==i & TL1_Peak$C0299==TRUE)
}
rm(i)
TL1CountPeak$ProbC02=TL1CountPeak$CountC0299/TL1CountPeak$TotalCount
for(i in 1:length(TL1CountPeak$bin)){ ## this counts total 99th perc CH4
reading in each bin
  TL1CountPeak$CountCH499[i]=sum(TL1_Peak$bin==i & TL1_Peak$CH499==TRUE)
}
rm(i)
TL1CountPeak$ProbCH4=TL1CountPeak$CountCH499/TL1CountPeak$TotalCount
TL1CountPeak$degree=seq(0,355,5)
TL1CountPeak$rad=TL1CountPeak$degree*pi/180
## ====Pollution Concentration Roses DF building
CountMeanPeakC02<-tapply(TL1_Peak$C02_dry, TL1_Peak$bin, mean) # this averages
the C02 concentration by bin
TL1CountPeak$AvgConcC02<-as.numeric(CountMeanPeakC02)
Count99MeanPeakC02<-by(TL1_Peak$C02_dry, list(TL1_Peak$bin, TL1_Peak$C0299),
mean) # this averages the C02 concentration by bin & 99%=TRUE))
for(i in 1:length(TL1CountPeak$bin)){
  TL1CountPeak$Avg99ConcC02[i]=Count99MeanPeakC02[i+72]
}
rm(i)
CountMeanPeakCH4<-tapply(TL1_Peak$CH4_dry, TL1_Peak$bin, mean) # this averages
the CH4 concentration by bin
TL1CountPeak$AvgConcCH4<-as.numeric(CountMeanPeakCH4)
Count99MeanPeakCH4<-by(TL1_Peak$CH4_dry, list(TL1_Peak$bin, TL1_Peak$CH499),
mean) # this averages the CH4 concentration by bin & 99%=TRUE))
for(i in 1:length(TL1CountPeak$bin)){
  TL1CountPeak$Avg99ConcCH4[i]=Count99MeanPeakCH4[i+72]
}
rm(i)
TL1CountPeak$deg_move<-TL1CountPeak$degree+2.5

ggplot(data=TL1_Peak, aes(TL1_Peak$DEG)) +geom_histogram(binwidth =
5,fill=I("navyblue")) +

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))
+
  coord_polar(theta="x", start=0, direction =1)+
  ggtitle("TL1 Oct.18, 2013, 4pm-7pm Wind Direction")+
  theme(plot.title = element_text(size=14, face="plain"))+

```

```

  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
  face="plain"), axis.title.y=element_text(face="plain",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

ggplot(data=TL1CountPeak, aes(x=TL1CountPeak$deg_move,
y=TL1CountPeak$ProbCO2))+
  geom_bar(stat="identity", fill="steelblue")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("TL1 99perc CO2 ProbRose 10/18 4pm-6pm")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
  face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

ggplot(data=TL1CountPeak, aes(x=TL1CountPeak$deg_move,
y=TL1CountPeak$ProbCH4))+
  geom_bar(stat="identity", fill="pink")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("TL1 99perc CH4 ProbRose 10/18 4pm-6pm")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
  face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

## ===== Expected & Residual Calculations ===== ###
Exp99<-0.01
TL1Count$Res99CO2<-(TL1Count$ProbCO2-Exp99CO2^2)/(Exp99CO2^2)
ggplot(data=TL1Count, aes(x=TL1Count$deg_move, y=TL1Count$Res99CO2))+
  geom_bar(stat="identity", fill="coral1")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("TL1 99perc CO2 Residual")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
  face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))
TL1Count$Res99CH4<-((TL1Count$ProbCH4-Exp99)^2)/(Exp99^2)

```

```

ggplot(data=TL1Count, aes(x=TL1Count$deg_move, y=TL1Count$Res99CH4))+
  geom_bar(stat="identity", fill="firebrick2")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("TL1 99perc CH4 Residual")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

###-----subset data, remove windspeed less than 1m/s ----###
TL1WS <- TL1[ which(TL1$WindSpeed >= 1), ]
quantile(TL1WS$CO2_dry, c(.99)) ##437.3469
percentileRose(TL1WS, pollutant="CO2_dry", percentile=99, method="cpf",
col="green", smooth=TRUE) #99th percentile rose
percentileRose(TL1WS, pollutant="CH4_dry", percentile=99, method="cpf",
col="darkorange", smooth=TRUE) #99th percentile rose
windRose(TL1WS)

#### ----- OPENAIR stuff -----#####
TL1$wd<-TL1$DEG
TL1$ws<-TL1$WindSpeed
TL1$date<-TL1$DT
local.tz="America/Denver"
latitude=40.7607800
longitude=-111.8910500
percentileRose(TL1, pollutant="CO2_dry", percentile=99, method="cpf",
col="green", smooth=TRUE) #99th percentile rose
percentileRose(TL1, pollutant="CO2_dry", percentile=99, method="cpf",
type="hour", col="green", smooth=TRUE) #hourly
percentileRose(subset(TL1, DT>"2013-10-17 09:04:00.000 MDT" & DT<="2013-10-17
11:00:00.000 MDT"), pollutant="CO2_dry", percentile=99, method="cpf",
col="green", smooth=TRUE) #hourly
percentileRose(subset(TL1, DT>"2013-10-17 14:00:00.000 MDT" & DT<="2013-10-17
16:00:00.000 MDT"), pollutant="CO2_dry", percentile=99, method="cpf",
col="green", smooth=TRUE) #hourly
percentileRose(subset(TL1, DT>"2013-10-18 14:00:00.000 MDT" & DT<="2013-10-18
15:00:00.000 MDT"), pollutant="CO2_dry", percentile=99, method="cpf",
col="green", smooth=TRUE) #hourly
percentileRose(subset(TL1, DT>"2013-10-17 16:00:00.000 MDT" & DT<="2013-10-17
20:00:00.000 MDT"), pollutant="CO2_dry", percentile=99, method="cpf",
col="green", smooth=TRUE) #hourly
percentileRose(subset(TL1, DT>"2013-10-17 20:00:00.000 MDT" & DT<="2013-10-18
00:00:00.000 MDT"), pollutant="CO2_dry", percentile=99, method="cpf",
col="green", smooth=TRUE) #hourly
###---note: need to write a loop that will look at every two hours

polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=99, main="TL1:
99% CH4 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=99, main="TL1:
99% CO2 (ppm)")

```

```

polarPlot(TL1, pollutant="CO2_dry") #bivariate polar plot also incorporating
windspeed
### subsetting day and night data
DT1<-TL1$DT[1]
DT2<-"2013-10-17 18:45:00.000 MDT"
DT3<-"2013-10-18 07:40:00.000 MDT"
DT4<-TL1$DT[nrow(TL1)]
int <- interval(DT1, DT2)
int2<-interval(DT2, DT3)
int3<-interval(DT3, DT4)
TL1_Oct17day<-TL1[TL1$DT %within% int,]
TL1_night<-TL1[TL1$DT %within% int2,]
TL1_Oct18day<-TL1[TL1$DT %within% int3,]
TL1_day<-rbind(TL1_Oct17day,TL1_Oct18day)
rm(TL1_Oct17day,TL1_Oct18day)
TL1$Daylight<-
#--- Plotting with Open Air
percentileRose(TL1_day, pollutant="CO2_dry", percentile=99, method="cpf",
col="green", smooth=TRUE) #99th percentile rose
percentileRose(TL1_night, pollutant="CO2_dry", percentile=99, method="cpf",
col="blue", smooth=TRUE)
percentileRose(TL1_day, pollutant="CH4_dry", percentile=99, method="cpf",
col="darkorange", smooth=TRUE)
percentileRose(TL1_night, pollutant="CH4_dry", percentile=99, method="cpf",
col="darkred", smooth=TRUE)
percentileRose(TL1, pollutant="CO2_dry", percentile=99, method="cpf",
type="hour", col="green", smooth=TRUE) #hourly
polarPlot(TL1_day, pollutant="CO2_dry")
polarPlot(TL1_night, pollutant="CO2_dry")
polarPlot(TL1_day, pollutant="CH4_dry")
polarPlot(TL1_night, pollutant="CH4_dry")
polarPlot(TL1, pollutant="CO2_dry", uncertainty=TRUE)

polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=99, min.bin=3,
main="TL1: 99th Perc. CO2") # CPF bivariate plots
polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=99, min.bin=3,
main="TL1: 99th Perc. CH4") # CPF bivariate plots

polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(0,10),
main="TL1: 0%-10% CO2 (ppm)") # CPF bivariate plots
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(10,20),
main="TL1: 10%-20% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(20,30),
main="TL1: 20%-30% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(30,40),
main="TL1: 30%-40% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(40,50),
main="TL1: 40%-50% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(50,60),
main="TL1: 50%-60% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(60,70),
main="TL1: 60%-70% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(70,80),

```

```

main="TL1: 70%-80% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(80,90),
main="TL1: 80%-90% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(90,100),
main="TL1: 90%-100% CO2 (ppm)")

polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(0,20),
main="TL1: 0%-20% CO2 (ppm)") # CPF bivariate plots
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(20,40),
main="TL1: 20%-40% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(40,60),
main="TL1: 40%-60% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(60,80),
main="TL1: 60%-80% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(80,100),
main="TL1: 80%-100% CO2 (ppm)")

polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(0,25),
main="TL1: 0%-25% CO2 (ppm)") # CPF bivariate plots
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(25,50),
main="TL1: 25%-50% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(50,75),
main="TL1: 50%-75% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(75,100),
main="TL1: 75%-100% CO2 (ppm)")

polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(0,50),
main="TL1: 0%-50% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(50,100),
main="TL1: 50%-100% CO2 (ppm)")

polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(0,33),
main="TL1: 0%-33% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(33,66),
main="TL1: 33%-66% CO2 (ppm)")
polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=c(66,100),
main="TL1: 66%-100% CO2 (ppm)")

polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=c(0,20),
main="TL1: 0%-20% CH4 (ppm)") # CPF bivariate plots
polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=c(20,40),
main="TL1: 20%-40% CH4 (ppm)")
polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=c(40,60),
main="TL1: 40%-60% CH4 (ppm)")
polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=c(60,80),
main="TL1: 60%-80% CH4 (ppm)")
polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=c(80,100),
main="TL1: 80%-100% CH4 (ppm)")

polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=c(0,25),
main="TL1: 0%-25% CH4 (ppm)") # CPF bivariate plots
polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=c(25,50),
main="TL1: 25%-50% CH4 (ppm)")
polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=c(50,75),

```



```

main="TL1: 50%-75% CH4 (ppm)")
polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=c(75,100),
main="TL1: 75%-100% CH4 (ppm)")

polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=c(0,10),
main="TL1: 0%-10% CH4 (ppm)") # CPF bivariate plots
polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=c(10,20),
main="TL1: 10%-20% CH4 (ppm)")
polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=c(30,40),
main="TL1: 30%-40% CH4 (ppm)") # CPF bivariate plots
polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=c(40,50),
main="TL1: 40%-50% CH4 (ppm)")

quantile(TL1$CO2_dry, probs = seq(0, 1, by = 0.1), na.rm = TRUE)
quantile(TL1$CH4_dry, probs = seq(0, 1, by = 0.1), na.rm = TRUE)

##### --- loop to create and save CBPF images in OpenAir --
setwd("~/Dropbox/Research/ProbRosePaper/Images/TL1_CPF")
plot_list = list()
percent<-c(0,20,40,60,80,100)
for(i in 1:(length(percent)-1)){
  j<-percent[i]
  k<-j+(percent[i+1]-percent[i])
  p=polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=c(j,k),
main=paste("TL1: ",j,"%-",k,"% CH4 (ppm)",collapse = '', sep = ''))
  plot_list[[i]] = p
  file_name =paste("TL1_CH4_",j,"to",k,".jpeg",collapse = '', sep = '')
#paste("iris_plot_", i, ".tiff", sep="")
  jpeg(file_name)
  print(plot_list[[i]])
  dev.off()
}

####--- compare night & day CBPF in OpenAir-----#####
# set tz to Utah (GMT - 6, but note the '+'; see openair manual)
latitude<-39.4
longitude<-111.9
TL1$date <-ymd_hms(TL1$DT, tz = "GMT")
#TL1$date <-ymd_hms(TL1$DT, tz = "Etc/GMT+6")
# divide data by daylight, nighttime
#TL1 <- cutData(TL1, type = "daylight") # doesn't work!!!
sunrise<-parse_date_time("7:00:00", orders="hms")
sunset<-parse_date_time("19:00:10", orders="hms")
TL1$daylight<-ifelse(
  hour(TL1$date)>=hour(sunrise) & hour(TL1$date)<hour(sunset), "daylight",
"nighttime"
)
windRose(TL1, type="daylight", main="TL1 WindRose Day vs. Night")
# percentile for daylight
percentileRose(subset(TL1, daylight == "daylight"), poll= "CO2_dry",
percentile=99, method="cpf", col="grey", smooth=TRUE, main="TL1: Day 99th
Perc. CO2")
polarPlot(subset(TL1, daylight == "daylight"), poll= "CO2_dry", stati="cpf",
min.bin=3, percentile = 99, main="TL1: Day CO2 99%")

```

```

polarPlot(subset(TL1, daylight == "daylight"), poll= "CO2_dry", stati="cpf",
percentile = c(90, 100), main="TL1: Day CO2 90%-100%")
polarPlot(subset(TL1, daylight == "daylight"), poll= "CO2_dry", stati="cpf",
percentile = c(0, 25), main="TL1: Day CO2 0%-25%")
polarPlot(subset(TL1, daylight == "daylight"), poll= "CO2_dry", stati="cpf",
percentile = c(80, 90), main="TL1: Day CO2 80%-90%")
polarPlot(subset(TL1, daylight == "daylight"), poll= "CO2_dry", stati="cpf",
percentile = c(80, 90), main="TL1: Day CO2 80%-90%")
polarPlot(subset(TL1, daylight == "daylight"), poll= "CO2_dry", stati="cpf",
percentile = c(60, 70), main="TL1: Day CO2 60%-70%")
plot_list = list()
percent<-c(0,20,40,60,80,100)
for(i in 1:(length(percent)-1)){
  j<-percent[i]
  k<-j+(percent[i+1]-percent[i])
  p=polarPlot(subset(TL1, daylight == "daylight"), pollutant="CH4_dry",
statistic="cpf", percentile=c(j,k), main=paste("TL1: Day ",j,"%-",k,"% CH4
(ppm)",collapse = '', sep = ''))
  plot_list[[i]] = p
  file_name =paste("TL1_CH4_Day_",j,"to",k,".jpeg",collapse = '', sep = '')
  jpeg(file_name)
  print(plot_list[[i]])
  dev.off()
}
plot_list = list()
percent<-c(0,10,20,30,40,50,60,70,80,90,100)
for(i in 1:(length(percent)-1)){
  j<-percent[i]
  k<-j+(percent[i+1]-percent[i])
  p=polarPlot(subset(TL1, daylight == "daylight"), pollutant="CH4_dry",
statistic="cpf", percentile=c(j,k), main=paste("TL1: Day ",j,"%-",k,"% CH4
(ppm)",collapse = '', sep = ''))
  plot_list[[i]] = p
  file_name =paste("TL1_CH4_Day_",j,"to",k,".jpeg",collapse = '', sep = '')
  #paste("iris_plot_", i, ".tiff", sep="")
  jpeg(file_name)
  print(plot_list[[i]])
  dev.off()
}
plot_list = list()
percent<-c(0,20,40,60,80,100)
for(i in 1:(length(percent)-1)){
  j<-percent[i]
  k<-j+(percent[i+1]-percent[i])
  p=polarPlot(subset(TL1, daylight == "daylight"), pollutant="CO2_dry",
statistic="cpf", percentile=c(j,k), main=paste("TL1: Day ",j,"%-",k,"% CO2
(ppm)",collapse = '', sep = ''))
  plot_list[[i]] = p
  file_name =paste("TL1_CO2_Day_",j,"to",k,".jpeg",collapse = '', sep = '')
  #paste("iris_plot_", i, ".tiff", sep="")
  jpeg(file_name)
  print(plot_list[[i]])
  dev.off()
}
plot_list = list()

```

```

percent<-c(0,10,20,30,40,50,60,70,80,90,100)
for(i in 1:(length(percent)-1)){
  j<-percent[i]
  k<-j+(percent[i+1]-percent[i])
  p=polarPlot(subset(TL1, daylight == "daylight"), pollutant="CO2_dry",
statistic="cpf", percentile=c(j,k), main=paste("TL1: Day ",j,"%-",k,"% CO2
(ppm)",collapse = '', sep = ''))
  plot_list[[i]] = p
  file_name =paste("TL1_CO2_Day_",j,"to",k,".jpeg",collapse = '', sep = '')
#paste("iris_plot_", i, ".tiff", sep="")
  jpeg(file_name)
  print(plot_list[[i]])
  dev.off()
}
# percentile for nighttime
percentileRose(subset(TL1, daylight == "nighttime"), poll= "CO2_dry",
percentile=99, method="cpf", col="grey", smooth=TRUE, main="TL1: Night 99th
Perc. CO2")
polarPlot(subset(TL1, daylight == "nighttime"), poll= "CO2_dry", stati="cpf",
min.bin=3, percentile = 99, main="TL1: Night CO2 99%")
polarPlot(subset(TL1, daylight == "nighttime"), poll= "CO2_dry", stati="cpf",
percentile = c(90, 100), main="TL1: Night CO2 90%-100%")
polarPlot(subset(TL1, daylight == "nighttime"), poll= "CO2_dry", stati="cpf",
percentile = c(0, 25), main="TL1: Night CH4 0%-25%")
# percentile for daylight CH4
percentileRose(subset(TL1, daylight == "daylight"), poll= "CH4_dry",
percentile=99, method="cpf", col="grey", smooth=TRUE, main="TL1: Day 99th
Perc. CH4")
polarPlot(subset(TL1, daylight == "daylight"), poll= "CH4_dry", stati="cpf",
min.bin=3, percentile = 99, main="TL1: Day CH4 99%")
polarPlot(subset(TL1, daylight == "daylight"), poll= "CH4_dry", stati="cpf",
percentile = c(90, 100), main="TL1: Day CH4 90%-100%")
polarPlot(subset(TL1, daylight == "daylight"), poll= "CH4_dry", stati="cpf",
percentile = c(0, 25), main="TL1: Day CH4 0%-25%")
# percentile for nighttime CH4
percentileRose(subset(TL1, daylight == "nighttime"), poll= "CH4_dry",
percentile=99, method="cpf", col="grey", smooth=TRUE, main="TL1: Night 99th
Perc. CH4")
polarPlot(subset(TL1, daylight == "nighttime"), poll= "CH4_dry", stati="cpf",
min.bin=3, percentile = 99, main="TL1: Night CH4 99%")
polarPlot(subset(TL1, daylight == "nighttime"), poll= "CH4_dry", stati="cpf",
percentile = c(90, 100), main="TL1: Night CH4 90%-100%")
polarPlot(subset(TL1, daylight == "nighttime"), poll= "CH4_dry", stati="cpf",
percentile = c(0, 25), main="TL1: Night CH4 0%-25%")

plot_list = list()
percent<-c(0,20,40,60,80,100)
for(i in 1:(length(percent)-1)){
  j<-percent[i]
  k<-j+(percent[i+1]-percent[i])
  p=polarPlot(subset(TL1, daylight == "nighttime"), pollutant="CH4_dry",
statistic="cpf", percentile=c(j,k), main=paste("TL1: Night ",j,"%-",k,"% CH4
(ppm)",collapse = '', sep = ''))
  plot_list[[i]] = p
  file_name =paste("TL1_CH4_Night_",j,"to",k,".jpeg",collapse = '', sep = '')

```

```

#paste("iris_plot_", i, ".tiff", sep="")
  jpeg(file_name)
  print(plot_list[[i]])
  dev.off()
}
plot_list = list()
percent<-c(0,10,20,30,40,50,60,70,80,90,100)
for(i in 1:(length(percent)-1)){
  j<-percent[i]
  k<-j+(percent[i+1]-percent[i])
  p=polarPlot(subset(TL1, daylight == "nighttime"), pollutant="CH4_dry",
  statistic="cpf", percentile=c(j,k), main=paste("TL1: Night ",j,"%-",k,"% CH4
(ppm)",collapse = '', sep = ''))
  plot_list[[i]] = p
  file_name =paste("TL1_CH4_Night_",j,"to",k,".jpeg",collapse = '', sep = '')
#paste("iris_plot_", i, ".tiff", sep="")
  jpeg(file_name)
  print(plot_list[[i]])
  dev.off()
}
plot_list = list()
percent<-c(0,20,40,60,80,100)
for(i in 1:(length(percent)-1)){
  j<-percent[i]
  k<-j+(percent[i+1]-percent[i])
  p=polarPlot(subset(TL1, daylight == "nighttime"), pollutant="CO2_dry",
  statistic="cpf", percentile=c(j,k), main=paste("TL1: Night ",j,"%-",k,"% CO2
(ppm)",collapse = '', sep = ''))
  plot_list[[i]] = p
  file_name =paste("TL1_CO2_Night_",j,"to",k,".jpeg",collapse = '', sep = '')
#paste("iris_plot_", i, ".tiff", sep="")
  jpeg(file_name)
  print(plot_list[[i]])
  dev.off()
}
plot_list = list()
percent<-c(0,10,20,30,40,50,60,70,80,90,100)
for(i in 1:(length(percent)-1)){
  j<-percent[i]
  k<-j+(percent[i+1]-percent[i])
  p=polarPlot(subset(TL1, daylight == "nighttime"), pollutant="CO2_dry",
  statistic="cpf", percentile=c(j,k), main=paste("TL1: Night ",j,"%-",k,"% CO2
(ppm)",collapse = '', sep = ''))
  plot_list[[i]] = p
  file_name =paste("TL1_CO2_Night_",j,"to",k,".jpeg",collapse = '', sep = '')
#paste("iris_plot_", i, ".tiff", sep="")
  jpeg(file_name)
  print(plot_list[[i]])
  dev.off()
}

polarPlot(TL1, pollutant="CH4_dry", statistic="cpf", percentile=99, main="TL1:
99% CH4 (ppm)")
polarPlot(subset(TL1, daylight == "nighttime"), pollutant="CH4_dry",
statistic="cpf", percentile=99, main="TL1: Nighttime 99% CH4 (ppm)")

```

```

polarPlot(subset(TL1, daylight == "nighttime"), pollutant="CO2_dry",
statistic="cpf", percentile=99, main="TL1: Nighttime 99% CO2 (ppm)")

polarPlot(subset(TL1, daylight == "daylight"), pollutant="CH4_dry",
statistic="cpf", percentile=99, main="TL1: Daytime 99% CH4 (ppm)")
polarPlot(subset(TL1, daylight == "daylight"), pollutant="CO2_dry",
statistic="cpf", percentile=99, main="TL1: Daytime 99% CO2 (ppm)")

polarPlot(TL1, pollutant="CO2_dry", statistic="cpf", percentile=99, main="TL1:
99% CO2 (ppm)")

##### ----- Subset on time/Rush Hour ----- ###
#TL1 10/18/2013 wsa a friday, evening rush btwn 4:00-6:00pm
MorningRushStart1<-as.POSIXct(strptime("2013-10-17 9:00:00", "%Y-%m-%d
%H:%M:%S"))
MorningRushEnd1<-as.POSIXct(strptime("2013-10-17 10:00:00", "%Y-%m-%d
%H:%M:%S"))
MorningRushStart2<-as.POSIXct(strptime("2013-10-18 7:00:00", "%Y-%m-%d
%H:%M:%S"))
MorningRushEnd2<-as.POSIXct(strptime("2013-10-18 10:00:00", "%Y-%m-%d
%H:%M:%S"))
Oct17AMRH_1<-subset(TL1, DT>MorningRushStart1 & DT<MorningRushEnd1)
Oct17AMRH_2<-subset(TL1, DT>MorningRushStart2 & DT<MorningRushEnd2)
Oct17AMRH<-rbind(Oct17AMRH_1, Oct17AMRH_2)

percentileRose(Oct17AMRH, pollutant="CO2_dry", percentile=99, method="cpf",
col="grey", smooth=TRUE, main="TL1: 99th Perc. CPF CO2 AM Rush-hour") #99th
percentile rose
percentileRose(Oct17AMRH, pollutant="CH4_dry", percentile=99, method="cpf",
col="grey", smooth=TRUE, main="TL1: 99th Perc. CPF CH4 AM Rush-hour") #99th
percentile rose
windRose(Oct17AMRH, main="TL1: Wind Rose AM Rush Hour")
polarPlot(Oct17AMRH, pollutant="CO2_dry", statistic="cpf", percentile=99,
main="TL1: 99th Perc. CO2 AM Rush Hour")
polarPlot(Oct17AMRH, pollutant="CH4_dry", statistic="cpf", percentile=99,
main="TL1: 99th Perc. CH4 AM Rush Hour")

EveRushStart<-as.POSIXct(strptime("2013-10-17 16:00:00", "%Y-%m-%d %H:%M:%S"))
EveRushEnd<-as.POSIXct(strptime("2013-10-17 18:00:00", "%Y-%m-%d %H:%M:%S"))
Oct17PMRH<-subset(TL1, DT>EveRushStart & DT<EveRushEnd)
percentileRose(Oct17PMRH, pollutant="CO2_dry", percentile=99, method="cpf",
col="grey", smooth=TRUE, main="TL1: 99th Perc. CPF CO2 PM Rush-hour") #99th
percentile rose
percentileRose(Oct17PMRH, pollutant="CH4_dry", percentile=99, method="cpf",
col="grey", smooth=TRUE, main="TL1: 99th Perc. CPF CH4 PM Rush-hour") #99th
percentile rose
windRose(Oct17PMRH, main="TL1: Wind Rose PM Rush Hour") #Note only for
10/17/13
polarPlot(Oct17PMRH, pollutant="CO2_dry", statistic="cpf", percentile=99,
min.bin=3, main="TL1: 99th Perc. CO2 PM Rush Hour")
polarPlot(Oct17PMRH, pollutant="CH4_dry", statistic="cpf", percentile=99,
min.bin=3, main="TL1: 99th Perc. CH4 PM Rush Hour")
polarPlot(Oct17AMRH, pollutant="CO2_dry", statistic="cpf", percentile=99,
min.bin=3, main="TL1: 99th Perc. CO2 AM Rush Hour")
polarPlot(Oct17AMRH, pollutant="CH4_dry", statistic="cpf", percentile=99,

```

```

min.bin=3, main="TL1: 99th Perc. CH4 AM Rush Hour")
polarPlot(Oct17AMRH, pollutant="CO2_dry", statistic="cpf",
percentile=c(90,100), min.bin=3, main="TL1: 90%-100% CO2 AM Rush Hour
10/17/13")
polarPlot(Oct17AMRH, pollutant="CH4_dry", statistic="cpf",
percentile=c(90,100), min.bin=3, main="TL1: 90%-100% CH4 AM Rush Hour
10/17/13")

##### ----- Subset on early Friday Morning ----- ###
#TL1 10/18/2013 was a friday, campus "gets going at 5am-7am
# This is to see if there is a diff direction from 5-7 then from 7-10
MorningRushStart3<-as.POSIXct(strptime("2013-10-18 5:00:00", "%Y-%m-%d
%H:%M:%S"))
MorningRushEnd3<-as.POSIXct(strptime("2013-10-18 7:00:00", "%Y-%m-%d
%H:%M:%S"))
Oct18AMRH_3<-subset(TL1, DT>MorningRushStart3 & DT<MorningRushEnd3)
percentileRose(Oct18AMRH_3, pollutant="CO2_dry", percentile=99, method="cpf",
col="grey", smooth=TRUE, main="TL1: 99th Perc. CPF CO2 5AM-7AM") #99th
percentile rose
windRose(Oct18AMRH_3, main="TL1: Wind Rose 5AM-7AM 10/18/13")
polarPlot(Oct18AMRH_3, pollutant="CO2_dry", statistic="cpf", percentile=99,
min.bin=3, main="TL1: 99th Perc. CO2 5AM-7AM 10/18/13")
polarPlot(Oct18AMRH_3, pollutant="CH4_dry", statistic="cpf", percentile=99,
min.bin=3, main="TL1: 99th Perc. CH4 5AM-7AM 10/18/13")
# This is to see if there is a diff direction from 4-6 then from 7-10
MorningRushStart4<-as.POSIXct(strptime("2013-10-18 4:00:00", "%Y-%m-%d
%H:%M:%S"))
MorningRushEnd4<-as.POSIXct(strptime("2013-10-18 7:00:00", "%Y-%m-%d
%H:%M:%S"))
Oct18AMRH_4<-subset(TL1, DT>MorningRushStart4 & DT<MorningRushEnd4)
percentileRose(Oct18AMRH_4, pollutant="CO2_dry", percentile=99, method="cpf",
col="grey", smooth=TRUE, main="TL1: 99th Perc. CPF CO2 4AM-7AM") #99th
percentile rose
windRose(Oct18AMRH_4, main="TL1: Wind Rose 4AM-7AM 10/18/13")
polarPlot(Oct18AMRH_4, pollutant="CO2_dry", statistic="cpf", percentile=99,
min.bin=1, main="TL1: 99th Perc. CO2 4AM-7AM 10/18/13")
polarPlot(Oct18AMRH_4, pollutant="CH4_dry", statistic="cpf", percentile=99,
min.bin=3, main="TL1: 99th Perc. CH4 4AM-7AM 10/18/13")

#####
TL1$hour<-hour(TL1$DT)
TL1$ws_group<-as.integer(TL1$ws)
for(i in 0:max(TL1$ws_group)){
  data<-subset(TL1,ws_group==i)
  title<-paste("Cor",i,sep='')
  assign(title, cor(data[,c(4,6,7,23,24,27,28)]))
  #write.table(title, file=paste("Cor",i,".csv",sep=''),sep=",")
}
write.table(Cor0, file="Cor0.csv",sep=",")
write.table(Cor1, file="Cor1.csv",sep=",")
write.table(Cor2, file="Cor2.csv",sep=",")
write.table(Cor3, file="Cor3.csv",sep=",")
write.table(Cor4, file="Cor4.csv",sep=",")
write.table(Cor5, file="Cor5.csv",sep=",")
write.table(Cor6, file="Cor6.csv",sep=",")

```

A.2 FASB Script

This script take the data from the Picarro © CRDS raw data files for site FASB1, formats the data, accounts for lag time, performs statistics, and creates the CBPF plots.

```
setwd("~/Dropbox/Research/MinckData/FASB1_Oct18_19")
#setwd("C:/Users/Meg/Dropbox/Research/MinckData/FASB1_Oct18_19")
list.files(path = ".", pattern = NULL, all.files = FALSE, full.names = FALSE,
recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALSE)
FASB1<-read.table("~/Dropbox/Research/MinckData/Raw/FASB_HP20131018.txt",
header=TRUE, sep = ',')

names(FASB1)
class(FASB1)
str(FASB1)
class(FASB1$DATE) #Returns type of class of that column
FASB1$DATE[1] #returns first date
tail(FASB1, 10) #prints last 10 rows
head(FASB1, 20) # CHECK first lines for any 0 data!!!
FASB1 <-FASB1[-c(1), ] #removes first row

## Time date stuff
library(lubridate)
# ==== create date time column
op <- options(digits.secs = 3) ##Need to set this
FASB1$DT<-as.POSIXct(strftime(paste(FASB1$DATE, FASB1$TIME), format="%Y/%m/%d
%H:%M:%OS3"), tz="GMT")
library(lubridate)
FASB1$DT<-with_tz(FASB1$DT, tzzone="America/Denver")
hour(FASB1$DT[1]) # returns just hour of date-timestamp

start<-FASB1$DT[1]
end<-FASB1$DT[length(FASB1$DT)]
span<-interval(start,end)
as.duration(span)
as.period(span, unit="hours")

## WIND Calculations
FASB1$WindSpeed=sqrt(FASB1$ANEMOMETER_UX^2 + FASB1$ANEMOMETER_UY^2 +
FASB1$ANEMOMETER_UZ^2) #units=m/s
FASB1$N = FASB1$ANEMOMETER_UX;
FASB1$E = FASB1$ANEMOMETER_UY;
FASB1$D = (180/pi)*atan2(FASB1$E,FASB1$N); #converted to degrees
##Need to get rid of negative values
##### instead of If statements, use this!
idx1 <- FASB1$D < 0
idx2 <- FASB1$D > 0
idx3 <- FASB1$D == 0
FASB1$D1[idx1] <- FASB1$D[idx1] * (-1)
FASB1$D1[idx2] <- (FASB1$D[idx2] * (-1)) + 360
```

```

FASB1$D1[idx3] <- 0

OFFNDEG<-0 #offset angle for anemometer at FASB1
idx4 <- (360-FASB1$D1)<=OFFNDEG
idx5 <- (360-FASB1$D1)>OFFNDEG
FASB1$DEG[idx4] <- (FASB1$D1[idx4]-360+OFFNDEG)
FASB1$DEG[idx5] <- (FASB1$D1[idx5]+OFFNDEG)
any(is.na(FASB1$DEG)) # make sure there are no NA's
write.table(FASB1, "FASB1_windDir.txt", sep=",") #write table because
sometimes Rstudio crashes

#===== Graphics =====
hist(FASB1$DEG, 72, main="Wind Direction, FASB1, Oct. 18-19, 2013",
col="grey",xlab="Direction (Degrees)")
hist(FASB1$CO2_dry, breaks=50, main="CO2 dry, FASB1, Oct. 18-19, 2013",
col="grey",xlab="CO2 dry (ppmv)")
abline(v=quantile(FASB1$CO2_dry, c(.99)), col="red", lwd=3) #adds 99th
percentile
hist(FASB1$CH4_dry, breaks=60, main="CH4 dry, FASB1, Oct. 18-19, 2013",
col="grey",xlab="CH4 dry (ppmv)")
abline(v=quantile(FASB1$CH4_dry, c(.99)), col="red", lwd=3) #adds 99th
percentile
hist(FASB1$WindSpeed, col="grey", main="FASB1 Wind Speed", xlab="Wind Speed
(m/s)")
hist(FASB1$WindSpeed[FASB1$CO299==TRUE], col="grey", main="FASB1 Wind Speed
for 99th percentile CO2 readings", xlab="Wind Speed (m/s)")
hist(FASB1$WindSpeed[FASB1$CH499==TRUE], col="grey", main="FASB1 Wind Speed
for 99th percentile CH4 readings", xlab="Wind Speed (m/s)")

FASB199CO2<-quantile(FASB1$CO2_dry, c(.99))
FASB199CH4<-quantile(FASB1$CH4_dry, c(.99))

# Circular plot
WindDir<-circular(FASB1$DEG)
rose.diag(WindDir, type="angles", units = 'degrees', zero = pi/2) ## Rotates
where 0deg is

## Statistics
summary(FASB1$CO2_dry)
summary(FASB1$CH4_dry)
summary(FASB1$WindSpeed)
quantile(FASB1$CO2_dry, c(.99))
quantile(FASB1$CH4_dry, c(.99))

## Standard Deviation
sd(FASB1$CO2_dry)
sd(FASB1$CH4_dry)
sd(FASB1$WindSpeed)

## ===== Plotting ===== ##
# Note: dev.off() clears plotting window
boxplot(FASB1$CO2_dry, cex=16, ylab="CO2 Dry (ppmv)", main="CO2dry FASB1, Oct.
18-19, 2013")

```



```

boxplot(FASB1$CH4_dry, cex=16, ylab="CH4 Dry (ppmv)", main="CO2dry FASB1, Oct.
18-19, 2013")
plot(FASB1$CO2_dry, type='h')
## HISTOGRAMS
hist(FASB1$CO2_dry)
hist(FASB1$CO2_dry, breaks=75, cex=16, xlab="CO2 Dry (ppmv)", main="CO2dry
FASB1, Oct. 18-19, 2013")
boxplot(FASB1$CH4_dry, cex.axis=1.4, ylab="CH4 Dry (ppmv)", main="CO2dry
FASB1, Oct. 18-19, 2013")
## ===== Timeseries
library(xts)
zoo<-zoo(FASB1$CO2_dry,FASB1$DT) #use zoo becuase they are not constant time
intervals
plot(zoo, main = "CO2dry FASB1, Oct. 18-19, 2013", ylab="Co2dry (ppmv)",
xlab=NULL) # this needs some formatting
abline(h=FASB199CO2, col="red", lwd=3) #adds 99th percentile

plot(zoo, xaxt = "n",main = "CO2dry FASB1, Oct. 18-19, 2013", ylab="Co2dry
(ppmv)", xlab="")
axis(1, FASB1$DT, format(FASB1$DT, "%d %T"), las=2, cex.axis = 1)
# ==== try combining both into one plot
FASB1_TS<-cbind(FASB1$CO2_dry, FASB1$CH4_dry, FASB1$WindSpeed)
FASB1_TSzoo<-zoo(FASB1_TS, FASB1$DT)
summary(FASB1_TSzoo)
plot(FASB1_TSzoo, plot.type="multiple", main="FASB1 Oct. 18-19, 2013",
cex.main=1.5, xlab="", ylab=list("CO2 dry(ppmv)", "CH4 dry(ppmv)", "Wind Speed
(m/s)"))
axis(1, FASB1$DT, format(FASB1$DT, "%d %T"), las=2, cex.axis = 1)

# ===== SImple statistics/Modeling =====
#remove some of the unused variables, or find a way to designate certain
paramters
cor(FASB199CO2)
cov(FASB199CO2)
cor.test(FASB199CO2$CO2_dry, FASB199CO2$DEG, method="spearman")
cor.test(FASB199CO2$CO2_dry, FASB199CO2$WindSpeed, method="spearman")
plot(FASB199CO2)
colnames(FASB199CH4) #Columns names
pairs(FASB199CH4[,c(4,6,7,13,18)], main="99th perc. CH4dry FASB1") # sepcific
columns
pairs(~CH4_dry+CO2_dry+H2O+DEG+WindSpeed, data=FASB199CH4, main="99th perc.
CH4dry FASB1") # sepcific columns
pairs(~CH4_dry+CO2_dry+H2O+DEG+WindSpeed, data=FASB199CO2, main="99th perc.
CO2dry FASB1") # sepcific columns

#===== Circular Statistics =====
library(circular)
FASB1circ<-circular(FASB1$DEG, units = 'degrees', zero = pi/2)
mean(FASB1circ)
MeanResultLength<-rho.circular(FASB1circ)
CircSD_FASB1<-sd.circular(FASB1circ)
V<-(1-MeanResultLength) #Variance

# ===== Linear-Circular Association

```

```

FASB1_Radians<-FASB1$DEG*2*pi/360
R2xtIndTestRand(FASB1$CO2_dry,FASB1_Radians,9999) #Johnson-Wehrly-Mardia
Corelation COefficient

# ===== Linear-Circular Association Mardia's rank coefficient
uniscor<-OrderScores(FASB1$CO2_dry,FASB1_Radians)
MardiaRankIndTestRand(uniscor, 9999)

# ===== Rayleigh Test of significance of mean direction (pg 81 Pewsey)
rtest<-rayleigh.test(FASB1$circ)
rtest

#===Subset dat a to 99th percentile CO2
CO2dry99<-quantile(FASB1$CO2_dry, c(.99))
FASB199CO2 <- FASB1[ which(FASB1$CO2_dry >= CO2dry99), ]
FASB199CO2$circ<-circular(FASB199CO2$DEG, units = 'degrees', zero = pi/2,
rotation="counter")
mean(FASB199CO2$circ)
MeanResultLengthFASB199CO2<-rho.circular(FASB199CO2$circ)
CircSD_FASB199CO2<-sd.circular(FASB199CO2$circ)
V_FASB199CO2<-(1-MeanResultLengthFASB199CO2) #Variance
FASB199CO2_Rad<-FASB199CO2$DEG*2*pi/360
# ===== Linear-Circular Association Mardia's rank coefficient
uniscor99CO2<-OrderScores(FASB199CO2$CO2_dry,FASB199CO2_Rad)
MardiaRankIndTestRand(uniscor99CO2, 9999)

#===Subset dat a to 99th percentile CH4
CH4dry99<-quantile(FASB1$CH4_dry, c(.99))
FASB199CH4 <- FASB1[ which(FASB1$CH4_dry >= CH4dry99), ]
FASB199CH4$circ<-circular(FASB199CH4$DEG, units = 'degrees', zero = pi/2,
rotation="counter")
mean(FASB199CH4$circ)
MeanResultLengthFASB199CH4<-rho.circular(FASB199CH4$circ)
CircSD_FASB199CH4<-sd.circular(FASB199CH4$circ)
V_FASB199CH4<-(1-MeanResultLengthFASB199CH4) #Variance
FASB199CH4_Rad<-FASB199CH4$DEG*2*pi/360
# ===== Linear-Circular Association Mardia's rank coefficient
uniscor99CH4<-OrderScores(FASB199CH4$CH4_dry,FASB199CH4_Rad)
MardiaRankIndTestRand(uniscor99CH4, 9999)
# ===== Rayleigh Test of significance of mean direction (pg 81 Pewsey)
rtestCO2<-rayleigh.test(FASB199CO2$circ) #CO2
rtestCH4<-rayleigh.test(FASB199CH4$circ) #CH4

# ===== Prob ROse Plots and Polar Histograms =====#
#===Breaking up degrees into bins
FASB1$bin<-trunc(0.2*FASB1$DEG)+1
FASB1$CO299<-FASB1$CO2_dry >= CO2dry99 #If CO2 is >= 99th percentile, returns
a true
FASB1$CH499<-FASB1$CH4_dry >= CH4dry99 #If CH4 is >= 99th percentile, returns
a true

#===== Create new dataset with 72 rows for count data
FASB1Count<-data.frame("bin"=numeric(72), "TotalCount"=numeric(72),
"CountCO299"=numeric(72), "ProbCO2"=numeric(72),"AvgConcCO2"=numeric(72))

```

```

FASB1Count$bin=seq(1,72)
for(i in 1:length(FASB1Count$bin)){ ##this counts total readings in each bin
  FASB1Count$TotalCount[i]=sum(FASB1$bin==i)
}
rm(i)
for(i in 1:length(FASB1Count$bin)){ ## this counts total 99th perc CO2
  reading in each bin
  FASB1Count$CountCO299[i]=sum(FASB1$bin==i & FASB1$CO299==TRUE)
}
rm(i)
FASB1Count$ProbCO2=FASB1Count$CountCO299/FASB1Count$TotalCount

for(i in 1:length(FASB1Count$bin)){ ## this counts total 99th perc CH4
  reading in each bin
  FASB1Count$CountCH499[i]=sum(FASB1$bin==i & FASB1$CH499==TRUE)
}
rm(i)
FASB1Count$ProbCH4=FASB1Count$CountCH499/FASB1Count$TotalCount

FASB1Count$degree=seq(0,355,5)
FASB1Count$rad=FASB1Count$degree*pi/180
plot(FASB1Count$degree, FASB1Count$ProbCO2, pch=18, col='red', cex=1.4,
main='FASB1 Oct. 18-19, 2013', xlab='Degrees', ylab='99th Perentile CO2
Probability', font.lab=2)

## =====Pollution Concentration Roses
CountMeanCO2<-tapply(FASB1$CO2_dry, FASB1$bin, mean) # this averages the CO2
concentration by bin
FASB1Count$AvgConcCO2<-as.numeric(CountMeanCO2)
# -or-
#for(i in 1:length(FASB1Count$bin)){
#  FASB1Count$AvgConcCO2[i]=CountMeanCO2[i]
#}
# -or-
#CountMeanCO2<-by(FASB1$CO2_dry, FASB1$bin, mean)
#FASB1Count$AvgConcCO2<-CountMeanCO2
plot(FASB1Count$degree, FASB1Count$AvgConcCO2, pch=18, col='red', xlab='Bin
Number', ylab='FASB1 Avg CO2 Pollution', font.lab=2)

Count99MeanCO2<-by(FASB1$CO2_dry, list(FASB1$bin, FASB1$CO299), mean) # this
averages the CO2 concentration by bin & 99%=TRUE))
for(i in 1:length(FASB1Count$bin)){
  FASB1Count$Avg99ConcCO2[i]=Count99MeanCO2[i+72]
}
rm(i)
plot(FASB1Count$degree, FASB1Count$Avg99ConcCO2, pch=18, col='red', xlab='Bin
Number', ylab='FASB1 Avg 99th Perentile CO2 Pollution', font.lab=2)

CountMeanCH4<-tapply(FASB1$CH4_dry, FASB1$bin, mean) # this averages the CH4
concentration by bin
FASB1Count$AvgConcCH4<-as.numeric(CountMeanCH4)
Count99MeanCH4<-by(FASB1$CH4_dry, list(FASB1$bin, FASB1$CH499), mean) # this
averages the CH4 concentration by bin & 99%=TRUE))
for(i in 1:length(FASB1Count$bin)){
  FASB1Count$Avg99ConcCH4[i]=Count99MeanCH4[i+72]
}

```

```

}
rm(i)
#=== Conventional Pollution Plots
library(plotrix)
polar.plot(FASB1Count$AvgConcCO2,FASB1Count$degree,main="FASB1, CO2 Pollution
Conc Rose", radial.lim=c(390,394),start=90,clockwise=TRUE,lwd=4,line.col=1)
polar.plot(FASB1Count$Avg99ConcCO2,FASB1Count$degree,main="FASB1, 99th % CO2
Pollution Conc Rose",
radial.lim=c(415,425),start=90,clockwise=TRUE,lwd=4,line.col=4)
polar.plot(FASB1Count$AvgConcCH4,FASB1Count$degree,main="FASB1, CH4 Pollution
Conc Rose", radial.lim=c(1.7,1.72),start=90,clockwise=TRUE,lwd=4,line.col=4)
polar.plot(FASB1Count$Avg99ConcCH4,FASB1Count$degree,main="FASB1, 99th % CH4
Pollution Conc Rose",
radial.lim=c(1.787,1.795),start=90,clockwise=TRUE,lwd=4,line.col=4)

# == Wind Frequency
TotWind<-sum(FASB1Count$TotalCount)
FASB1Count$WindFreq<-FASB1Count$TotalCount/TotWind
FASB1Count$CO299WindFreq<-FASB1Count$ProbCO2/FASB1Count$WindFreq
FASB1Count$CH499WindFreq<-FASB1Count$ProbCH4/FASB1Count$WindFreq
oldpar<-polar.plot(FASB1Count$CO299WindFreq,FASB1Count$degree,main="FASB1 CO2
Prob Rose/AvgWindFreq",
radial.lim=c(0,1.25),start=90,clockwise=TRUE,lwd=4,line.col=3)

### ===== CONDITIONAL PROBABILITY -> P(99th|ni) , Jun 2016 ===== #####
### This calculates the probability that there is a 99th percentile reading in
### a given wind direction bin using conditional probability trees
FASB1Count$Cond99CO2=FASB1Count$ProbCO2*FASB1Count$WindFreq
## the prob that there is a 99th % reading AND it is in a given bin
FASB1Count$Cond99CH4=FASB1Count$ProbCH4*FASB1Count$WindFreq

## ===== CIRC STATS ON 99TH PERC CO2 PROB DATA ===== ###
FASB1Countcirc<-circular(FASB1Count$degree, units = 'degrees', zero = pi/2,
rotation="counter")
mean(FASB1Countcirc)
MeanResultLengthFASB1Count<-rho.circular(FASB1Countcirc)
CircSD_FASB1Countcirc<-sd.circular(FASB1Countcirc)
V_FASB1Countcirc<-(1-MeanResultLengthFASB1Count) #Variance
# =Rayleigh Test of significance of mean direction (pg 81 Pewsey)
rtest<-rayleigh.test(FASB1Countcirc)
rtest

#Polar Plot/Probability Rose:
library(plotrix)
oldpar<-polar.plot(FASB1Count$ProbCO2,FASB1Count$degree,main="FASB1 Oct. 18-
19, 2013, CO2dry Prob Rose", radial.lim=c(0,.025),radial.labels=c(0,0.005,
0.01,
0.015,0.02,""),start=90,clockwise=TRUE,lwd=4,line.col=4,show.grid.labels=3)
oldpar<-radial.plot(FASB1Count$ProbCO2,FASB1Count$rad,main="FASB1 Oct. 18-19,
2013, CO2dry Prob Rose",
radial.lim=c(0,.025),start=1.5708,show.grid.labels=3,clockwise=TRUE,lwd=4,line
.col=4)
# Double check polar plot
axis(side = 1, font = 2) # x axis labels bold

```

```

par(font.axis = 2)
hist(FASB1$DEG, 72, main="Wind Direction, FASB1 Oct. 18-19, 2013",
col="grey",xlab="Direction (Degrees)", xlim=c(0,360), font.lab=2)
polarhist<-polar.plot(FASB1Count$TotalCount,FASB1Count$degree, main="Test: FWU
6/3/2015, Wind Hist",
radial.lim=c(0,150000),start=90,clockwise=TRUE,lwd=4,line.col=4)

# Check with OpenAir
windRose(FASB1, ws="WindSpeed", wd="DEG", breaks =c(0,.5,1,1.5,2,4,6))
pollutionRose(FASB1, pollutant="CO2_dry", ws="WindSpeed", wd="DEG",
breaks=c(380,390,400,410,420))
windRose(FASB199CO2, ws="WindSpeed", wd="DEG", breaks =c(0,.5,1,1.5,2,4,6))
pollutionRose(FASB199CO2, pollutant="CO2_dry", ws="WindSpeed", wd="DEG",
breaks=c(416,418,420,422,425,426))

# ggplot2 wind rose diagram!!!! ==== This works well!====
library(ggplot2)
ggplot(data=FASB1, aes(FASB1$DEG)) +geom_histogram(binwidth =
5,fill=I("blue")) +
scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))
+
coord_polar(theta="x", start=0, direction =1)+
ggtitle("FASB1 Oct. 18-19, 2013 Wind Direction")+
theme(plot.title = element_text(size=14, face="plain"))+
labs(x="", y="") + theme_bw() +
theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="plain", size=16),
panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))
# ====try with transparent background =====#
p<-ggplot(data=FASB1, aes(FASB1$DEG)) +geom_histogram(binwidth =
5,fill=I("black")) +

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))
+
  coord_polar(theta="x", start=0, direction =1)+
  ggtitle("FASB1 Oct. 18-19, 2013 Wind Direction")+
  theme(plot.title = element_text(size=14, face="plain"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="plain", size=16),
panel.background = element_rect(fill = "transparent",colour = NA),
  panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"),plot.background =
element_rect(fill = "transparent",colour = NA))
png('FASB1_WindHist.png',width=350,height=350,units="px",bg = "transparent")
print(p)

# ggplot2 wind rose diagram!!!! ==== 99th percentile wind histogram
library(ggplot2)
ggplot(data=FASB199CO2, aes(FASB199CO2$DEG)) +geom_histogram(binwidth =
5,fill=I("palegreen4")) +

```

```

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))
+
  coord_polar(theta="x", start=0, direction =1)+
  ggtitle("FASB1 Oct. 18-19, 2013 99perc CO2 Wind Direction")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="bold"), axis.text.x=element_text(size=14,
face="bold"), axis.title.y=element_text(face="bold", size=16),panel.grid.minor
= element_line(colour="grey"), panel.grid.major=element_line(colour = "grey"))
# == wind histogram of just 99th percentile data for CH4
ggplot(data=FASB199CH4, aes(FASB199CH4$DEG)) +geom_histogram(binwidth =
5,fill=I("yellowgreen")) +

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))
+
  coord_polar(theta="x", start=0, direction =1)+
  ggtitle("FASB1 Oct. 18-19, 2013 99th percentile CH4 Wind Direction")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="bold"), axis.text.x=element_text(size=14,
face="bold"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

# =====GGplot for Count Data, better Probability Rose diagrams
FASB1Count$deg_move<-FASB1Count$degree+2.5 # had to move degrees by +2.5 to
get this to work
ggplot(data=FASB1Count, aes(x=FASB1Count$deg_move, y=FASB1Count$ProbCO2))+
  geom_bar(stat="identity", fill="red")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("FASB1 99perc CO2 ProbRose")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))
ggplot(data=FASB1Count, aes(x=FASB1Count$deg_move, y=FASB1Count$ProbCH4))+
  geom_bar(stat="identity", fill="pink")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("FASB1 test 99perc CH4 ProbRose")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),

```

```

panel.grid.major=element_line(colour = "grey"))
## === Conditional Prob Plots
ggplot(data=FASB1Count, aes(x=FASB1Count$deg_move, y=FASB1Count$Cond99CO2))+
  geom_bar(stat="identity", fill="pink")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("FASB1 Conditional Prob CO2")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))
ggplot(data=FASB1Count, aes(x=FASB1Count$deg_move, y=FASB1Count$Cond99CH4))+
  geom_bar(stat="identity", fill="orange")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("FASB1 Conditional Prob CH4")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

# === ggplot prob rose/wind freq
ggplot(data=Mar6Count, aes(x=Mar6Count$deg_move, y=Mar6Count$CO299WindFreq))+
  geom_bar(stat="identity", fill="darkturquoise")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("3/6/14 99perc CO2 ProbRose/WindFreq")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))
ggplot(data=Mar6Count, aes(x=Mar6Count$deg_move, y=Mar6Count$CH499WindFreq))+
  geom_bar(stat="identity", fill="purple")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("3/6/14 99perc CH4 ProbRose/WindFreq")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),

```

```
panel.grid.major=element_line(colour = "grey"))
```

```
### ===== PEAK TIME PR
===== #####
# =====Subset dataframe based on time ===== #
PeakStart<-as.POSIXct(strptime("2013-10-19 14:00:00", "%Y-%m-%d %H:%M:%S"))
PeakEnd<-as.POSIXct(strptime("2013-10-19 16:00:00", "%Y-%m-%d %H:%M:%S"))
FASB1_Peak<-subset(FASB1, DT>PeakStart & DT<PeakEnd)
#===== Create new dataset with 72 rows for count data
FASB1CountPeak<-data.frame("bin"=numeric(72), "TotalCount"=numeric(72),
"CountCO299"=numeric(72), "ProbCO2"=numeric(72),"AvgConcCO2"=numeric(72))
FASB1CountPeak$bin=seq(1,72)
for(i in 1:length(FASB1CountPeak$bin)){ ##this counts total readings in each
bin
  FASB1CountPeak$TotalCount[i]=sum(FASB1_Peak$bin==i)
}
rm(i)
for(i in 1:length(FASB1CountPeak$bin)){ ## this counts total 99th perc CO2
reading in each bin
  FASB1CountPeak$CountCO299[i]=sum(FASB1_Peak$bin==i & FASB1_Peak$CO299==TRUE)
}
rm(i)
FASB1CountPeak$ProbCO2=FASB1CountPeak$CountCO299/FASB1CountPeak$TotalCount
for(i in 1:length(FASB1CountPeak$bin)){ ## this counts total 99th perc CH4
reading in each bin
  FASB1CountPeak$CountCH499[i]=sum(FASB1_Peak$bin==i & FASB1_Peak$CH499==TRUE)
}
rm(i)
FASB1CountPeak$ProbCH4=FASB1CountPeak$CountCH499/FASB1CountPeak$TotalCount
FASB1CountPeak$degree=seq(0,355,5)
FASB1CountPeak$rad=FASB1CountPeak$degree*pi/180
## =====Pollution COncentration Roses DF building
CountMeanPeakCO2<-tapply(FASB1_Peak$CO2_dry, FASB1_Peak$bin, mean) # this
averages the CO2 concentration by bin
FASB1CountPeak$AvgConcCO2<-as.numeric(CountMeanPeakCO2)
Count99MeanPeakCO2<-by(FASB1_Peak$CO2_dry, list(FASB1_Peak$bin,
FASB1_Peak$CO299), mean) # this averages the CO2 concentration by bin &
99%=TRUE))
for(i in 1:length(FASB1CountPeak$bin)){
  FASB1CountPeak$Avg99ConcCO2[i]=Count99MeanPeakCO2[i+72]
}
rm(i)
CountMeanPeakCH4<-tapply(FASB1_Peak$CH4_dry, FASB1_Peak$bin, mean) # this
averages the CH4 concentration by bin
FASB1CountPeak$AvgConcCH4<-as.numeric(CountMeanPeakCH4)
Count99MeanPeakCH4<-by(FASB1_Peak$CH4_dry, list(FASB1_Peak$bin,
FASB1_Peak$CH499), mean) # this averages the CH4 concentration by bin &
99%=TRUE))
for(i in 1:length(FASB1CountPeak$bin)){
  FASB1CountPeak$Avg99ConcCH4[i]=Count99MeanPeakCH4[i+72]
}
rm(i)
FASB1CountPeak$deg_move<-FASB1CountPeak$degree+2.5
```



```

ggplot(data=FASB1_Peak, aes(FASB1_Peak$DEG)) +geom_histogram(binwidth =
5,fill=I("navyblue")) +

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))
+
  coord_polar(theta="x", start=0, direction =1)+
  ggtitle("FASB1 Oct.19, 2013, 2pm-4pm Wind Direction")+
  theme(plot.title = element_text(size=14, face="plain"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
  face="plain"), axis.title.y=element_text(face="plain",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

ggplot(data=FASB1CountPeak, aes(x=FASB1CountPeak$deg_move,
y=FASB1CountPeak$ProbCO2))+
  geom_bar(stat="identity", fill="steelblue")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("FASB1 99perc CO2 ProbRose 10/19 2pm-4pm")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
  face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

ggplot(data=FASB1CountPeak, aes(x=FASB1CountPeak$deg_move,
y=FASB1CountPeak$ProbCH4))+
  geom_bar(stat="identity", fill="pink")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("FASB1 99perc CH4 ProbRose 10/19 2pm-4pm")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
  face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

#### ----- OPENAIR stuff -----#####
FASB1$wd<- FASB1$DEG
FASB1$ws<- FASB1$WindSpeed
FASB1$date<- FASB1$DT
local.tz="America/Denver"
local.tz="America/Denver"
latitude=40.7607800
longitude=-111.8910500
percentileRose(FASB1, pollutant="CO2_dry", percentile=99, method="cpf",

```

```

col="green", smooth=TRUE) #99th percentile rose
percentileRose(FASB1, pollutant="CO2_dry", percentile=99, method="cpf",
type="hour", col="green", smooth=TRUE) #hourly

polarPlot(FASB1, pollutant="CO2_dry")
polarPlot(FASB1, pollutant="CO2_dry", statistic="cpf", percentile=99) # CPF
bivariate plots
polarPlot(FASB1, pollutant="CO2_dry", statistic="cpf", percentile=c(0,10)) #
CPF bivariate plots
polarPlot(FASB1, pollutant="CO2_dry", statistic="cpf", percentile=c(10,20))
polarPlot(FASB1, pollutant="CO2_dry", statistic="cpf", percentile=c(75,100))
polarPlot(FASB1, pollutant="CO2_dry", statistic="cpf", percentile=c(25,50))
polarPlot(FASB1, pollutant="CO2_dry", statistic="cpf", percentile=c(50,75))
polarPlot(FASB1, pollutant="CO2_dry", statistic="cpf", percentile=c(90,99.99))
polarPlot(FASB1, pollutant="CO2_dry", statistic="cpf", percentile=c(80,90))
polarPlot(FASB1, pollutant="CH4_dry", statistic="cpf", percentile=99) # CPF
bivariate plots

##### --- loop to create and save CBPF images in OpenAir --
#####
setwd("~/Dropbox/Research/ProbRosePaper/Images/FASB1_CPF")
plot_list = list()
percent<-c(0,20,40,60,80,100)
for(i in 1:(length(percent)-1)){
  j<-percent[i]
  k<-j+(percent[i+1]-percent[i])
  p=polarPlot(FASB1, pollutant="CH4_dry", statistic="cpf", percentile=c(j,k),
main=paste("FASB1: ",j,"%-",k,"% CH4 (ppm)",collapse = '', sep = ''))
  plot_list[[i]] = p
  file_name =paste("FASB1_CH4_",j,"to",k,".jpeg",collapse = '', sep = '')
#paste("iris_plot_", i, ".tiff", sep="")
  jpeg(file_name)
  print(plot_list[[i]])
  dev.off()
}
plot_list = list()
percent<-c(0,25,50,75,100)
for(i in 1:(length(percent)-1)){
  j<-percent[i]
  k<-j+(percent[i+1]-percent[i])
  p=polarPlot(FASB1, pollutant="CH4_dry", statistic="cpf", percentile=c(j,k),
main=paste("FASB1: ",j,"%-",k,"% CH4 (ppm)",collapse = '', sep = ''))
  plot_list[[i]] = p
  file_name =paste("FASB1_CH4_",j,"to",k,".jpeg",collapse = '', sep = '')
#paste("iris_plot_", i, ".tiff", sep="")
  jpeg(file_name)
  print(plot_list[[i]])
  dev.off()
}

####--- compare night & day CBPF in OpenAir-----####
# set tz to Utah (GMT - 6, but note the '+'; see openair manual)
latitude<-39.4
longitude<--111.9
#FASB1$date <-as.POSIXct(strptime(paste(FASB1$DATE, FASB1$TIME),

```

```

format="%Y/%m/%d %H:%M:%OS3"), tz="Etc/GMT+6") #doesnt work

# divide data by daylight, nighttime
#FASB1 <- cutData(FASB1, type = "daylight") #Doesnt work
sunrise<-parse_date_time("7:00:00", orders="hms")
sunset<-parse_date_time("19:00:10", orders="hms")
FASB1$daylight<-ifelse(
  hour(FASB1$date)>=hour(sunrise) & hour(FASB1$date)<hour(sunset), "daylight",
  "nighttime"
)
windRose(FASB1, type="daylight", main="FASB1 WindRose Day vs. Night")
# percentile for daylight
polarPlot(subset(FASB1, daylight == "daylight"), poll= "CO2_dry", stati="cpf",
percentile = 99, main="FASB1: Day CO2 99%")
polarPlot(subset(FASB1, daylight == "daylight"), poll= "CO2_dry", stati="cpf",
percentile = c(90, 100), main="FASB1: Day CO2 90%-100%")
polarPlot(subset(FASB1, daylight == "daylight"), poll= "CO2_dry", stati="cpf",
percentile = c(0, 25), main="FASB1: Day CO2 0%-25%")
# percentile for nighttime
polarPlot(subset(FASB1, daylight == "nighttime"), poll= "CO2_dry",
stati="cpf", percentile = 99, main="FASB1: Night CO2 99%")
polarPlot(subset(FASB1, daylight == "nighttime"), poll= "CO2_dry",
stati="cpf", percentile = c(90, 100), main="FASB1: Night CO2 90%-100%")
polarPlot(subset(FASB1, daylight == "nighttime"), poll= "CO2_dry",
stati="cpf", percentile = c(0, 25), main="FASB1: Night CO2 0%-25%")
# percentile for daylight CH4
polarPlot(subset(FASB1, daylight == "daylight"), poll= "CH4_dry", stati="cpf",
percentile = 99, main="FASB1: Day CH4 99%")
polarPlot(subset(FASB1, daylight == "daylight"), poll= "CH4_dry", stati="cpf",
percentile = c(90, 100), main="FASB1: Day CH4 90%-100%")
polarPlot(subset(FASB1, daylight == "daylight"), poll= "CH4_dry", stati="cpf",
percentile = c(0, 25), main="FASB1: Day CH4 0%-25%")
# percentile for nighttime CH4
polarPlot(subset(FASB1, daylight == "nighttime"), poll= "CH4_dry",
stati="cpf", percentile = 99, main="FASB1: Night CH4 99%")
polarPlot(subset(FASB1, daylight == "nighttime"), poll= "CH4_dry",
stati="cpf", percentile = c(90, 100), main="FASB1: Night CH4 90%-100%")
polarPlot(subset(FASB1, daylight == "nighttime"), poll= "CH4_dry",
stati="cpf", percentile = c(0, 25), main="FASB1: Night CH4 0%-25%")

FASB1night<-subset(FASB1, daylight=="nighttime", c(date, CO2_dry, CH4_dry, ws,
wd))
plot_list = list()
percent<-c(0,10,20,30,40,50,60,70,80,90,100)
for(i in 1:(length(percent)-1)){
  j<-percent[i]
  k<-j+(percent[i+1]-percent[i])
  p=polarPlot(FASB1night, pollutant="CO2_dry", statistic="cpf",
percentile=c(j,k), main=paste("FASB1: Night ",j,"%-",k,"% CO2 (ppm)",collapse
= '', sep = ''))
  plot_list[[i]] = p
  file_name =paste("FASB1_CO2_Night_",j,"to",k,".jpeg",collapse = '', sep =
'') #paste("iris_plot_", i, ".tiff", sep="")
  jpeg(file_name)
  print(plot_list[[i]])
}

```

```

    dev.off()
}

##### ---- Subset on time ---- ###
#FASB1 10/18/2013 was a friday, evening rush btwn 4:00-6:00pm
MorningRushStart<-as.POSIXct(strptime("2013-10-18 16:00:00", "%Y-%m-%d
%H:%M:%S"))
MorningRushEnd<-as.POSIXct(strptime("2013-10-18 18:00:00", "%Y-%m-%d
%H:%M:%S"))
Oct18RH<-subset(FASB1, DT>MorningRushStart & DT<MorningRushEnd)
percentileRose(Oct18RH, pollutant="CO2_dry", percentile=99, method="cpf",
min.bin=3, col="grey", smooth=TRUE, main="FASB1: 99th Perc. CPF CO2 Rush Hour
10/18/13") #99th percentile rose
percentileRose(Oct18RH, pollutant="CH4_dry", percentile=99, method="cpf",
min.bin=3,col="grey", smooth=TRUE, main="FASB1: 99th Perc. CPF CH4 Rush Hour
10/18/13") #99th percentile rose
windRose(Oct18RH, main="FASB1: Wind Rose Rush Hour 10/18/13")
polarPlot(Oct18RH, pollutant="CO2_dry", statistic="cpf", percentile=99,
min.bin=3, main="FASB1: 99th Perc. CO2 Rush Hour 10/18/2013")
polarPlot(Oct18RH, pollutant="CH4_dry", statistic="cpf", percentile=99,
min.bin=3, main="FASB1: 99th Perc. CH4 Rush Hour 10/18/2013")

#FASB1 10/19/2013 was a saturday, campus probably doesn't have an increase at
5am-7am
# This is to see if there is a diff direction from 5-7 then from other times
MorningRushStart3<-as.POSIXct(strptime("2013-10-19 5:00:00", "%Y-%m-%d
%H:%M:%S"))
MorningRushEnd3<-as.POSIXct(strptime("2013-10-19 7:00:00", "%Y-%m-%d
%H:%M:%S"))
Oct19AMRH_3<-subset(FASB1, DT>MorningRushStart3 & DT<MorningRushEnd3)
polarPlot(Oct19AMRH_3, pollutant="CO2_dry", statistic="cpf", percentile=99,
min.bin=3, main="FASB1: 99th Perc. CO2 5AM-7AM 10/18/13")
polarPlot(Oct19AMRH_3, pollutant="CH4_dry", statistic="cpf", percentile=99,
min.bin=3, main="FASB1: 99th Perc. CH4 5AM-7AM 10/18/13")

## Subset on early days
# This is to see if there is a diff direction from 4-6 then from 7-10
MorningRushStart4<-as.POSIXct(strptime("2013-10-19 4:00:00", "%Y-%m-%d
%H:%M:%S"))
MorningRushEnd4<-as.POSIXct(strptime("2013-10-19 7:00:00", "%Y-%m-%d
%H:%M:%S"))
Oct18AMRH_4<-subset(FASB1, DT>MorningRushStart4 & DT<MorningRushEnd4)
percentileRose(Oct18AMRH_4, pollutant="CO2_dry", percentile=99, method="cpf",
col="grey", smooth=TRUE, main="FASB1: 99th Perc. CPF CO2 4AM-7AM") #99th
percentile rose
windRose(Oct18AMRH_4, main="FASB1: Wind Rose 4AM-7AM 10/19/13")
polarPlot(Oct18AMRH_4, pollutant="CO2_dry", statistic="cpf", percentile=99,
min.bin=3, main="FASB1: 99th Perc. CO2 4AM-7AM 10/19/13")
polarPlot(Oct18AMRH_4, pollutant="CH4_dry", statistic="cpf", percentile=99,
min.bin=3, main="FASB1: 99th Perc. CH4 4AM-7AM 10/19/13")

```

A.3 March FASB Scripts

This script take the data from the Picarro[®] CRDS raw data files for site FASB in March, formats the data, accounts for lag time, performs statistics, and creates the CBPF plots.

```
setwd("~/Dropbox/Research/MinckData/Mar06")
#On Server: setwd("/Volumes/mwalsh/Documents/MinckData/TLI_Oct17_18_LagFixed")
list.files(path = ".", pattern = NULL, all.files = FALSE, full.names = FALSE,
recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALSE)
Mar6<-read.csv("Mar6.csv", header=TRUE, sep = ',')

names(Mar6)
class(Mar6)
str(Mar6)
class(Mar6$DATE) #Returns type of class of that column
Mar6$DATE[1] #returns first date
tail(Mar6, 10) #prints last 10 rows
head(Mar6, 20) #prints first 20 rows
# === !! Note that the first 6445 rows don't have anemometer data!!!!
Mar6 <-Mar6[-c(1:6445), ] #This removes the first 6445 rows of dataset
# ===== Note that there is a strange peak of CO2 at 9:02, start data set at
9:04
Mar6<-subset(Mar6, DT>"2013-10-17 09:04:00.000 MDT")

## ===== Time date stuff =====
## Note that the Picarro time is in GMT.
## Also note that we need to adjust for a lag time of 27.7 seconds.
op <- options(digits.secs = 3) ##Need to set this
Mar6$DT<-as.POSIXct(strftime(paste(Mar6$DATE, Mar6$TIME), format="%Y/%m/%d
%H:%M:%OS3"), tz="GMT")
library(lubridate)
Mar6$DT<-with_tz(Mar6$DT, tzone="America/Denver")
hour(Mar6$DT[1]) # returns just hour of date-timestamp

# ==== create date time column
start<-Mar6$DT[1]
end<-Mar6$DT[length(Mar6$DT)]
span<-interval(start,end)
as.duration(span)
as.period(span, unit="hours")

##count how many times CH4dry or CO2dry = 0
Mar6$CH4dry==0
sum(Mar6$CH4dry == 0) #determines how many times 0 value occurs
which(Mar6$CH4dry==0) #Returns position of 0 values
Mar6$CO2dry==0
sum(Mar6$CO2dry == 0) #determines how many times 0 value occurs
```

```

which(Mar6$CO2dry==0) #Returns position of 0 values

## WIND Calculations
Mar6$WindSpeed=sqrt(Mar6$ANEMOMETERUX^2 + Mar6$ANEMOMETERUY^2 +
Mar6$ANEMOMETERUZ^2) #units=m/s
Mar6$N = Mar6$ANEMOMETERUX;
Mar6$E = Mar6$ANEMOMETERUY;
Mar6$D = (180/pi)*atan2(Mar6$E,Mar6$N); #converted to degrees
##Need to get rid of negative values
##### instead of I statements, use this!
idx1 <- Mar6$D < 0
idx2 <- Mar6$D > 0
idx3 <- Mar6$D == 0
Mar6$D1[idx1] <- Mar6$D[idx1] * (-1)
Mar6$D1[idx2] <- (Mar6$D[idx2] * (-1)) + 360
Mar6$D1[idx3] <- 0

OFFNDEG<-6 #offset angle for anemometer at Mar6
idx4 <- (360-Mar6$D1)<=OFFNDEG
idx5 <- (360-Mar6$D1)>OFFNDEG
Mar6$DEG[idx4] <- (Mar6$D1[idx4]-360+OFFNDEG)
Mar6$DEG[idx5] <- (Mar6$D1[idx5]+OFFNDEG)
any(is.na(Mar6$DEG)) # make sure there are no NA's
write.table(Mar6, "~/Dropbox/Research/MinckData/Mar06/Mar6.txt", sep=",") #
write table because sometimes Rstudio crashes

# ===== Graphics =====#
hist(Mar6$DEG, 72, main="Wind Direction, FASB2 Mar 6 2014",
col="grey",xlab="Direction (Degrees)")
hist(Mar6$CO2dry, breaks=50, main="FASB2 CO2 dry, Mar 6 2014",
col="grey",xlab="CO2 dry (ppmv)")
abline(v=quantile(Mar6$CO2dry, c(.99)), col="red", lwd=3) #adds 99th
percentile
hist(Mar6$CH4dry, breaks=60, main="FASB2 CH4 dry, Mar 6 2014",
col="grey",xlab="CH4 dry (ppmv)")
abline(v=quantile(Mar6$CH4dry, c(.99)), col="red", lwd=3) #adds 99th
percentile
hist(Mar6$WindSpeed, breaks=50, main="FASB2 CH4 dry, Mar 6 2014",
col="grey",xlab="Wind Speed (m/s)")
hist(Mar699CO2$WindSpeed, breaks=30, main="FASB2 99th CO2 dry WindSpeed, Mar6
2014", col="grey",xlab="Wind Speed (m/s)")
hist(Mar699CH4$WindSpeed, breaks=30, main="FASB2 99th CH4 dry WindSpeed, Mar6
2014", col="grey",xlab="Wind Speed (m/s)")

# Circular plot#
###try subestting data
sample<-Mar6[ sample(Mar6$DEG), round(0.0001*length(Mar6$DEG)), ]
round(0.0001 * nrow(mydf[mydf$gender == "F",])) #.01% of wind deg data
sample_deg<-sample(Mar6$DEG,1000,replace = FALSE)

WindDir<-circular(Mar6$DEG)
#rose.diag(WindDir, type="angles", units = 'degrees', zero = pi/2) ## Rotates
where 0deg is

```

```

## Statistics
summary(Mar6$CO2dry)
summary(Mar6$CH4dry)
summary(Mar6$WindSpeed)
quantile(Mar6$CO2dry, c(.99)) # 99th percentile
quantile(Mar6$CH4dry, c(.99)) # 99th percentile

sd(Mar6$CO2dry)
sd(Mar6$CH4dry)
sd(Mar6$WindSpeed)
var(Mar6$CO2dry)
var(Mar6$CH4dry)
var(Mar6$WindSpeed)

## Plotting
# Note: dev.off() clears plotting window
boxplot(Mar6$CO2dry, cex=16, ylab="CO2 Dry (ppmv)", main="6/1/2015 CO2dry",
outcex=1)
boxplot(Mar6$CH4dry, cex=16, ylab="CH4 Dry (ppmv) 6/1/2015", outcex=1)
plot(Mar6$CO2dry, type='h')
## HISTOGRAMS
hist(Mar6$CO2dry)
hist(Mar6$CO2dry, breaks=75, cex=16, xlab="CO2 Dry (ppmv)", main="6/1/2015
CO2dry")
boxplot(Mar6$CH4dry, cex.axis=1.4, ylab="CH4 Dry (ppmv) 6/1/2015")

#===== Circular Statistics =====
library(circular)
Mar6circ<-circular(Mar6$DEG, units = 'degrees', zero = pi/2,
rotation="counter")
mean(Mar6circ)
MeanResultLength<-rho.circular(Mar6circ)
CircSD_TLI<-sd.circular(Mar6circ)
V<-(1-MeanResultLength) #Variance

# ===== Linear-Circular Association
Mar6_Radians<-Mar6$DEG*2*pi/360
R2xtIndTestRand(Mar6$CO2dry,Mar6_Radians,9999) #Johnson-Wehrly-Mardia
Corelation COefficient
# [1] 0.002859866 0.000100000
uniscors<-OrderScores(Mar6$CO2dry,Mar6_Radians) #All CO2
MardiaRankIndTestRand(uniscors, 9999)
uniscors<-OrderScores(Mar6$CH4dry,Mar6_Radians) #All CH4
MardiaRankIndTestRand(uniscors, 9999)

#===Subset daa to 99th percentile CO2
CO2dry99<-quantile(Mar6$CO2dry, c(.99))
Mar699CO2 <- Mar6[ which(Mar6$CO2dry >= CO2dry99), ]
Mar699circ<-circular(Mar699CO2$DEG, units = 'degrees', zero = pi/2,
rotation="counter")
mean(Mar699circ)

```

```

MeanResultLength99<-rho.circular(Mar699circ)
CircSD_TLI99<-sd.circular(Mar699circ)
V_Mar699<-(1-MeanResultLength99) #Variance

# ===== Linear-Circular Association
Mar699_Radians<-Mar699C02$DEG*2*pi/360
R2xtIndTestRand(Mar699C02$C02dry,Mar699_Radians,9999) #Johnson-Wehrly-Mardia
Corelation
uniscors<-OrderScores(Mar699C02$C02dry,Mar699_Radians)
MardiaRankIndTestRand(uniscors, 9999)

# ===== Rayleigh Test of significance of mean direction (pg 81 Pewsey)
rtest<-rayleigh.test(Mar6circ)
rtest

#===Subset daa to 99th percentile CH4
CH4dry99<-quantile(Mar6$CH4dry, c(.99))
Mar699CH4 <- Mar6[ which(Mar6$CH4dry >= CH4dry99), ]
Mar699CH4circ<-circular(Mar699CH4$DEG, units = 'degrees', zero = pi/2,
rotation="counter")
mean(Mar699CH4circ)
MeanResultLength99CH4<-rho.circular(Mar699CH4circ)
CircSD_TLI99CH4<-sd.circular(Mar699CH4circ)
V_Mar699CH4<-(1-MeanResultLength99CH4) #Variance
# ===== Linear-Circular Association
Mar699CH4_Radians<-Mar699CH4$DEG*2*pi/360
uniscors<-OrderScores(Mar699CH4$CH4dry,Mar699CH4_Radians)
MardiaRankIndTestRand(uniscors, 9999)
# ===== Rayleigh Test of significance of mean direction (pg 81 Pewsey)
rtestC02<-rayleigh.test(Mar699circ) #C02
rtestCH4<-rayleigh.test(Mar699CH4circ) #CH4

## ===== Timeseries
library(xts)
zooMar6C02<-zoo(Mar6$C02dry,Mar6$DT) #use zoo becuase they are not constant
time intervals
plot(zooMar6C02, main = "C02dry Mar6, Oct. 17-18, 2013", ylab="C02dry (ppmv)",
xlab=NULL) # this needs some formatting
abline(h=C02dry99, col="red", lwd=3) #adds 99th percentile

plot(zoo, xaxt = "n",main = "C02dry Mar6, Oct. 17-18, 2013", ylab="C02dry
(ppmv)", xlab="")
axis(1, Mar6$DT, format(Mar6$DT, "%d %T"), las=2, cex.axis = .7)

zooWind<-zoo(Mar6$DEG,Mar6$DT) #use zoo becuase they are not constant time
intervals
plot(zooWind, main = "Wind Direction FASB2 Mar6, 2014", ylab="C02dry (ppmv)",
xlab=NULL)

zooMar6CH4<-zoo(Mar6$CH4dry,Mar6$DT) #use zoo becuase they are not constant
time intervals
plot(zooMar6CH4, main = "CH4dry Mar6, Oct. 17-18, 2013", ylab="CH4dry (ppmv)",
xlab=NULL) # this needs some formatting
abline(h=CH4dry99, col="red", lwd=3)

```



```

# ==== try combining both into one plot
Mar6_TS<-cbind(Mar6$CO2dry, Mar6$CH4dry, Mar6$WindSpeed)
Mar6_TSzoo<-zoo(Mar6_TS, Mar6$DT)
summary(Mar6_TSzoo)
plot(Mar6_TSzoo, plot.type="multiple", main="FASB2 3/6/2014",xlab="",
ylab=list("CO2 dry(ppmv)", "CH4 dry(ppmv)", "Wind Speed (m/s)"), cex.lab=1,
cex.main=1.5, cex.axis=1.2)

# ===== SImple statistics/Modeling =====
#remove some of the unused variables, or find a way to designate certain
paramters
cor(Mar699CO2)
cov(Mar699CO2)
cor.test(Mar699CO2$CO2dry, Mar699CO2$DEG, method="spearman")
cor.test(Mar699CO2$CO2dry, Mar699CO2$WindSpeed, method="spearman")
plot(Mar699CO2)
colnames(Mar699CH4) #Columns names
pairs(Mar699CH4[,c(4,6,7,13,18)], main="99th perc. CH4dry Mar6") # sepcific
columns

#=== Training testing datasets
set.seed(10000)
train_idx <- sample(1:nrow(Mar6),1000,replace=FALSE)
train <- Mar6[train_idx,] # select all these rows
test <- Mar6[-train_idx,] # select all but these rows

#=== Building CARTs
library(rpart)
library(tree)
library(randomForest)
Mar6CO2.rpart<-rpart(CO2dry~DEG+WindSpeed+TIME+CH4dry, data=train)
plot(Mar6CO2.rpart)
text(Mar6CO2.rpart, cex=0.75)
plotcp(Mar6CO2.rpart)
plot(predict(Mar6CO2.rpart), residuals(Mar6CO2.rpart), xlab="Fitted values",
ylab="Residuals")
abline(h=0)
PredMar6CO2=predict(Mar6CO2.rpart, test)
plot(PredMar6CO2)
# Try to get a table of observed vs predicted
table(Original = train$CO2dry,Predicted = predict(Mar6CO2.rpart))

#===== KRUSKAL-WALLIS TEST =====#
kruskal.test(CO2dry ~ DEG, data = Mar699CO2)
#data: CO2dry by DEG
#Kruskal-Wallis chi-squared = 36385, df = 35257, p-value = 1.301e-05

# ===== Prob R0se Plots and Polar Histograms =====#
#===Breaking up degrees into bins
Mar6$bin<-trunc(0.2*Mar6$DEG)+1

```

```

Mar6$C0299<-Mar6$C02dry >= C02dry99 #If C02 is >= 99th percentile, returns a
true
Mar6$CH499<-Mar6$CH4dry >= CH4dry99 #If C02 is >= 99th percentile, returns a
true
#===== Create new dataset with 72 rows for count data
Mar6Count<-data.frame("bin"=numeric(72), "TotalCount"=numeric(72),
"Count99"=numeric(72),"Avg99Conc"=numeric(72))
Mar6Count$bin=seq(1,72)
Mar6Count$degrees=seq(0,355,5)
for(i in 1:length(Mar6Count$bin)){ ##this counts total readings in each bin
  Mar6Count$TotalCount[i]=sum(Mar6$bin==i)
}
rm(i)
for(i in 1:length(Mar6Count$bin)){ ## this counts total 99th perc reading
in each bin
  Mar6Count$CountC0299[i]=sum(Mar6$bin==i & Mar6$C0299==TRUE)
}
rm(i)
Mar6Count$ProbC02=Mar6Count$CountC0299/Mar6Count$TotalCount

for(i in 1:length(Mar6Count$bin)){ ## this counts total 99th perc CH4
reading in each bin
  Mar6Count$CountCH499[i]=sum(Mar6$bin==i & Mar6$CH499==TRUE)
}
rm(i)
Mar6Count$ProbCH4=Mar6Count$CountCH499/Mar6Count$TotalCount

plot(Mar6Count$degrees, Mar6Count$ProbC02, pch=18, col='red', xlab='Bin
Number', ylab='99th Perentile C02 Probability', font.lab=2)
## == Polar Probability ROse Plot:
oldpar<-polar.plot(Mar6Count$Prob,Mar6Count$degree,main="Test: Mar6, C02dry
Prob Rose", radial.lim=c(0,.05),start=90,clockwise=TRUE,lwd=4,line.col=4)

## =====Pollution COncentration Roses
CountMeanC02<-tapply(Mar6$C02dry, Mar6$bin, mean) # this averages the C02
concentration by bin
Mar6Count$AvgConcC02<-as.numeric(CountMean)
plot(Mar6Count$degrees, Mar6Count$AvgConcC02, pch=18, col='red', xlab='Bin
Number', ylab='Avg C02 Pollution', font.lab=2)

Count99MeanC02<-by(Mar6$C02dry, list(Mar6$bin, Mar6$C0299), mean) # this
averages the C02 concentration by bin & 99%=TRUE))
for(i in 1:length(Mar6Count$bin)){
  Mar6Count$Avg99Conc[i]=Count99MeanC02[i+72]
}
rm(i)
plot(Mar6Count$degrees, Mar6Count$Avg99Conc, pch=18, col='red', xlab='Bin
Number', ylab='Avg 99th Perentile C02 Pollution', font.lab=2)

# == Wind Frequency
TotWind<-sum(Mar6Count$TotalCount)
Mar6Count$WindFreq<-Mar6Count$TotalCount/TotWind
Mar6Count$C0299WindFreq<-Mar6Count$ProbC02/Mar6Count$WindFreq
Mar6Count$CH499WindFreq<-Mar6Count$ProbCH4/Mar6Count$WindFreq
oldpar<-polar.plot(Mar6Count$C0299WindFreq,Mar6Count$degree,main="Mar6 C02

```

```

Prob Rose/AvgWindFreq",
radial.lim=c(0,2.5),start=90,clockwise=TRUE,lwd=4,line.col=3)

### ==== CONDITIONAL PROBABILITY -> P(99th|ni) , Jun 2016 ===== #####
### This calculates the probability that there is a 99th percentile reading in
### a given wind direction bin using conditional probability trees
Mar6Count$Cond99C02=Mar6Count$ProbC02*Mar6Count$WindFreq
## the prob that there is a 99th % reading AND it is in a given bin
Mar6Count$Cond99CH4=Mar6Count$ProbCH4*Mar6Count$WindFreq

# === OPENAIR
pollutionRose(Mar6, pollutant = "CO2dry", ws="WindSpeed", wd="DEG",
breaks=c(300,350,400,450,500))
pollutionRose(Mar699C02, pollutant = "CO2dry", ws="WindSpeed", wd="DEG",
breaks=c(450,460, 470,480,490,500))

#=== Conventional Pollution Plots
library(plotrix)
polar.plot(Mar6Count$AvgConc,Mar6Count$degree,main="Mar6, CO2dry Pollution
Conc Rose", radial.lim=c(395,405),start=90,clockwise=TRUE,lwd=4,line.col=4)
polar.plot(Mar6Count$Avg99Conc,Mar6Count$degree,main="Mar6, 99th % CO2dry
Pollution Conc Rose",
radial.lim=c(450,470),start=90,clockwise=TRUE,lwd=4,line.col=4)

## === Wind Histograms with GGplot2
library(ggplot2)
ggplot(data=Mar6, aes(Mar6$DEG)) +geom_histogram(binwidth = 5,fill=I("blue"))
+
scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))
+
coord_polar(theta="x", start=0, direction =1)+
ggtitle("FASB2 Mar 6 2014 Wind Direction")+
theme(plot.title = element_text(size=14, face="bold"))+
labs(x="", y="") + theme_bw() +
theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="plain", size=16),
panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))
# == wind histogram of just 99th percentile data for CO2
ggplot(data=Mar699C02, aes(Mar699C02$DEG)) +geom_histogram(binwidth =
5,fill=I("palegreen4")) +
scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))
+
coord_polar(theta="x", start=0, direction =1)+
ggtitle("FASB2 Mar 6 2014 99th percentile CO2 Wind Direction")+
theme(plot.title = element_text(size=14, face="bold"))+
labs(x="", y="") + theme_bw() +
theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="bold"), axis.text.x=element_text(size=14,
face="bold"), axis.title.y=element_text(face="plain", size=16),
panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))
# == wind histogram of just 99th percentile data for CH4

```

```

ggplot(data=Mar699CH4, aes(Mar699CH4$DEG)) +geom_histogram(binwidth =
5,fill=I("yellowgreen")) +

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))
+
  coord_polar(theta="x", start=0, direction =1)+
  ggtitle("FASB2 Mar 6 2014 99th percentile CH4 Wind Direction")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="bold"), axis.text.x=element_text(size=14,
  face="bold"), axis.title.y=element_text(face="bold", size=16),
  panel.grid.minor = element_line(colour="grey"),
  panel.grid.major=element_line(colour = "grey"))

# ==== ProbRose with ggplot2
Mar6Count$deg_move<-Mar6Count$degree+2.5 # had to move degrees by +2.5 to
get this to work
ggplot(data=Mar6Count, aes(x=Mar6Count$deg_move, y=Mar6Count$ProbCO2))+
  geom_bar(stat="identity", fill="red")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("FASB2 Mar 6 2014 99perc CO2 ProbRose")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
  face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
  panel.grid.major=element_line(colour = "grey"))
ggplot(data=Mar6Count, aes(x=Mar6Count$deg_move, y=Mar6Count$ProbCH4))+
  geom_bar(stat="identity", fill="pink")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("FASB2 Mar 6 2014 99perc CH4 ProbRose")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
  face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
  panel.grid.major=element_line(colour = "grey"))
## === Conditional Prob Plots
ggplot(data=Mar6Count, aes(x=Mar6Count$deg_move, y=Mar6Count$Cond99CO2))+
  geom_bar(stat="identity", fill="pink")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("FASB2 3/6/14 Conditional Prob CO2")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,

```

```

    face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))
ggplot(data=Mar6Count, aes(x=Mar6Count$deg_move, y=Mar6Count$Cond99CH4))+
  geom_bar(stat="identity", fill="orange")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("FASB2 3/6/14 Conditional Prob CH4")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

# === ggplot prob rose/wind freq
ggplot(data=Mar6Count, aes(x=Mar6Count$deg_move, y=Mar6Count$C0299WindFreq))+
  geom_bar(stat="identity", fill="darkturquoise")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("Mar6 99perc C02 ProbRose/WindFreq")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))
ggplot(data=Mar6Count, aes(x=Mar6Count$deg_move, y=Mar6Count$CH499WindFreq))+
  geom_bar(stat="identity", fill="purple")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("Mar6 99perc CH4 ProbRose/WindFreq")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

### ===== PEAK TIME PR
===== #####
# =====Subset dataframe based on time ===== #
PeakStart<-as.POSIXct(strptime("2013-10-18 04:00:00", "%Y-%m-%d %H:%M:%S"))
PeakEnd<-as.POSIXct(strptime("2013-10-18 07:00:00", "%Y-%m-%d %H:%M:%S"))
Mar6_Peak<-subset(Mar6, DT>PeakStart & DT<PeakEnd)
#===== Create new dataset with 72 rows for count data

```

```

Mar6CountPeak<-data.frame("bin"=numeric(72), "TotalCount"=numeric(72),
"CountCO299"=numeric(72), "ProbCO2"=numeric(72),"AvgConcCO2"=numeric(72))
Mar6CountPeak$bin=seq(1,72)
for(i in 1:length(Mar6CountPeak$bin)){ ##this counts total readings in each
bin
  Mar6CountPeak$TotalCount[i]=sum(Mar6_Peak$bin==i)
}
rm(i)
for(i in 1:length(Mar6CountPeak$bin)){ ## this counts total 99th perc CO2
reading in each bin
  Mar6CountPeak$CountCO299[i]=sum(Mar6_Peak$bin==i & Mar6_Peak$CO299==TRUE)
}
rm(i)
Mar6CountPeak$ProbCO2=Mar6CountPeak$CountCO299/Mar6CountPeak$TotalCount
for(i in 1:length(Mar6CountPeak$bin)){ ## this counts total 99th perc CH4
reading in each bin
  Mar6CountPeak$CountCH499[i]=sum(Mar6_Peak$bin==i & Mar6_Peak$CH499==TRUE)
}
rm(i)
Mar6CountPeak$ProbCH4=Mar6CountPeak$CountCH499/Mar6CountPeak$TotalCount
Mar6CountPeak$degree=seq(0,355,5)
Mar6CountPeak$rad=Mar6CountPeak$degree*pi/180
## ====Pollution COncentration Roses DF building
CountMeanPeakCO2<-tapply(Mar6_Peak$CO2dry, Mar6_Peak$bin, mean) # this
averages the CO2 concentration by bin
Mar6CountPeak$AvgConcCO2<-as.numeric(CountMeanPeakCO2)
Count99MeanPeakCO2<-by(Mar6_Peak$CO2dry, list(Mar6_Peak$bin, Mar6_Peak$CO299),
mean) # this averages the CO2 concentration by bin & 99%=TRUE))
for(i in 1:length(Mar6CountPeak$bin)){
  Mar6CountPeak$Avg99ConcCO2[i]=Count99MeanPeakCO2[i+72]
}
rm(i)
CountMeanPeakCH4<-tapply(Mar6_Peak$CH4dry, Mar6_Peak$bin, mean) # this
averages the CH4 concentration by bin
Mar6CountPeak$AvgConcCH4<-as.numeric(CountMeanPeakCH4)
Count99MeanPeakCH4<-by(Mar6_Peak$CH4dry, list(Mar6_Peak$bin, Mar6_Peak$CH499),
mean) # this averages the CH4 concentration by bin & 99%=TRUE))
for(i in 1:length(Mar6CountPeak$bin)){
  Mar6CountPeak$Avg99ConcCH4[i]=Count99MeanPeakCH4[i+72]
}
rm(i)
Mar6CountPeak$deg_move<-Mar6CountPeak$degree+2.5

ggplot(data=Mar6_Peak, aes(Mar6_Peak$DEG)) +geom_histogram(binwidth =
5,fill=I("navyblue")) +

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))
+
  coord_polar(theta="x", start=0, direction =1)+
  ggtitle("Mar6 Oct.18, 2013, 4pm-7pm Wind Direction")+
  theme(plot.title = element_text(size=14, face="plain"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="plain",

```

```

size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

ggplot(data=Mar6CountPeak, aes(x=Mar6CountPeak$deg_move,
y=Mar6CountPeak$ProbCO2))+
  geom_bar(stat="identity", fill="steelblue")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("Mar6 99perc CO2 ProbRose 10/18 4pm-6pm")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

ggplot(data=Mar6CountPeak, aes(x=Mar6CountPeak$deg_move,
y=Mar6CountPeak$ProbCH4))+
  geom_bar(stat="identity", fill="pink")+
  coord_polar(theta="x", start=0, direction =1)+

scale_x_continuous(breaks=c(0,45,90,135,180,225,270,315,360),limits=c(0,360))+
  ggtitle("Mar6 99perc CH4 ProbRose 10/18 4pm-6pm")+
  theme(plot.title = element_text(size=14, face="bold"))+
  labs(x="", y="") + theme_bw() +
  theme(panel.border = element_blank(),axis.text.y=element_text(size=14,
face="plain"), axis.text.x=element_text(size=14,
face="plain"), axis.title.y=element_text(face="bold",
size=16),panel.grid.minor = element_line(colour="grey"),
panel.grid.major=element_line(colour = "grey"))

#### Subset data ####
subMar6<-Mar6[, c(5,7,11,16,17)]
save(subMar6,file="subMar6.RData")

####-----OpenAir-----####
setwd("~/Dropbox/Research/ProbRosePaper/Images/Mar_CPF")
Mar$ws<-Mar$WindSpeed
Mar$WindSpeed<-NULL
Mar$wd<-Mar$DEG
Mar$DEG<-NULL

windRose(Mar, main="FASB1 March 2014 Wind Rose")
polarPlot(Mar, pollutant="CH4dry", statistic="cpf", min.bin=3, percentile=99,
main="FASB1 March 2014: 99% CH4 (ppm)")
polarPlot(Mar, pollutant="CO2dry", statistic="cpf", min.bin=3, percentile=99,
main="FASB1 March 2014: 99% CO2 (ppm)")

polarPlot(Mar, pollutant="CO2dry", statistic="cpf", percentile=c(0,10),
main="FASB1 March: 0%-10% CO2 (ppm)") # CPF bivariate plots
polarPlot(Mar, pollutant="CO2dry", statistic="cpf", percentile=c(10,20),
main="FASB1 March: 10%-20% CO2 (ppm)")

```

```

polarPlot(Mar, pollutant="CO2dry", statistic="cpf", percentile=c(20,30),
main="FASB1 March: 20%-30% CO2 (ppm)")
polarPlot(Mar, pollutant="CO2dry", statistic="cpf", percentile=c(30,40),
main="FASB1 March: 30%-40% CO2 (ppm)")
polarPlot(Mar, pollutant="CO2dry", statistic="cpf", percentile=c(40,50),
main="FASB1 March: 40%-50% CO2 (ppm)")
polarPlot(Mar, pollutant="CO2dry", statistic="cpf", percentile=c(50,60),
main="FASB1 March: 50%-60% CO2 (ppm)")
polarPlot(Mar, pollutant="CO2dry", statistic="cpf", percentile=c(60,70),
main="FASB1 March: 60%-70% CO2 (ppm)")
polarPlot(Mar, pollutant="CO2dry", statistic="cpf", percentile=c(70,80),
main="FASB1 March: 70%-80% CO2 (ppm)")
polarPlot(Mar, pollutant="CO2dry", statistic="cpf", percentile=c(80,90),
main="FASB1 March: 80%-90% CO2 (ppm)")
polarPlot(Mar, pollutant="CO2dry", statistic="cpf", percentile=c(90,100),
main="FASB1 March: 90%-100% CO2 (ppm)")

polarPlot(Mar, pollutant="CH4dry", statistic="cpf", percentile=c(0,10),
main="FASB1 March: 0%-10% CH4 (ppm)") # CPF bivariate plots
polarPlot(Mar, pollutant="CH4dry", statistic="cpf", percentile=c(10,20),
main="FASB1 March: 10%-20% CH4 (ppm)")
polarPlot(Mar, pollutant="CH4dry", statistic="cpf", percentile=c(20,30),
main="FASB1 March: 20%-30% CH4 (ppm)")
polarPlot(Mar, pollutant="CH4dry", statistic="cpf", percentile=c(30,40),
main="FASB1 March: 30%-40% CH4 (ppm)")
polarPlot(Mar, pollutant="CH4dry", statistic="cpf", percentile=c(40,50),
main="FASB1 March: 40%-50% CH4 (ppm)")
polarPlot(Mar, pollutant="CH4dry", statistic="cpf", percentile=c(50,60),
main="FASB1 March: 50%-60% CH4 (ppm)")
polarPlot(Mar, pollutant="CH4dry", statistic="cpf", percentile=c(60,70),
main="FASB1 March: 60%-70% CH4 (ppm)")
polarPlot(Mar, pollutant="CH4dry", statistic="cpf", percentile=c(70,80),
main="FASB1 March: 70%-80% CH4 (ppm)")
polarPlot(Mar, pollutant="CH4dry", statistic="cpf", percentile=c(80,90),
main="FASB1 March: 80%-90% CH4 (ppm)")
polarPlot(Mar, pollutant="CH4dry", statistic="cpf", percentile=c(90,100),
main="FASB1 March: 90%-100% CH4 (ppm)")

polarPlot(Mar, pollutant="CH4dry", statistic="cpf", percentile=c(0,25),
main="FASB1 March: 0%-25% CH4 (ppm)")
polarPlot(Mar, pollutant="CH4dry", statistic="cpf", percentile=c(25,50),
main="FASB1 March: 25%-50% CH4 (ppm)")
polarPlot(Mar, pollutant="CH4dry", statistic="cpf", percentile=c(50,75),
main="FASB1 March: 50%-75% CH4 (ppm)")
polarPlot(Mar, pollutant="CH4dry", statistic="cpf", percentile=c(75,100),
main="FASB1 March: 75%-100% CH4 (ppm)")

plot_list = list()
percent<-c(0,20,40,60,80,100)
for(i in 1:(length(percent)-1)){
  j<-percent[i]
  k<-j+(percent[i+1]-percent[i])
  p=polarPlot(Mar, pollutant="CH4dry", statistic="cpf", percentile=c(j,k),
main=paste("FASB1 March: ",j,"%-",k,"% CH4 (ppm)",collapse = '', sep = ''))
  plot_list[[i]] = p
}

```



```

    file_name =paste("Mar_CH4_",j,"to",k,".jpeg",collapse = '', sep = '')
    #paste("iris_plot_", i, ".tiff", sep="")
    jpeg(file_name)
    print(plot_list[[i]])
    dev.off()
}

save(Mar, file="FASB2_Mar2014.RData")

##### ---- Time seies stuff
# Make column of just time
Mar6$Time<-format(Mar6$DT, "%H:%M:%OS3")
Mar7$Time<-format(Mar7$DT, "%H:%M:%OS3")
Mar8$Time<-format(Mar8$DT, "%H:%M:%OS3")
#####----get important columns
Mar8<-subset(Mar8, select=c(Time,CO2dry,CH4dry, WindSpeed, DEG))
Mar8$wd<-Mar8$DEG
Mar8$ws<-Mar8$WindSpeed
Mar8$WindSpeed<-NULL
Mar8$DEG<-NULL
Mar7<-subset(Mar7, select=c(Time,CO2dry,CH4dry, ws, wd))
Mar6<-subset(Mar6, select=c(Time,CO2dry,CH4dry, WindSpeed, DEG))
Mar6$wd<-Mar6$DEG
Mar6$ws<-Mar6$WindSpeed
Mar6$WindSpeed<-NULL
Mar6$DEG<-NULL
##### Combine all three
March <- cbind(Mar6,Mar7,Mar8)

```

APPENDIX B

UNIVERSITY OF UTAH UNION ROOF CASE STUDY METHODOLOGY

B.1 Study Background

For the Union building roof research study (Figure B.1), an LI-7500A Open Path CO₂/H₂O Analyzer, LI-7700 Open Path CH₄ Analyzer, Licor Smartflux[®] onsite processing unit and data-logger, and a Gill HS-50 three-dimensional (3D) anemometer are being used. The instruments were attached to an adjustable tripod tower (see Figure B.2). In addition to CO₂, CH₄, H₂O, and wind data, the system also collects temperature and pressure from both the Smartflux[®] unit and the LI-7700 unit. The Licor tower was set up on the University of Utah Ray Olpin Student Union building in June of 2016. Instruments have been continuously collecting CO₂, CH₄, H₂O, and wind data since that time. The Union building roof was used as the study site because it is expected that the exhaust outlets from the restaurant kitchens would be characterized by a higher CO₂ and CH₄ signal during hours of operation (approximately 9 am to 4 pm).

The overall objective of this study is to use statistical tests and visual plots to discern probable greenhouse gas emission source locations using high frequency CO₂, CH₄, and meteorological data. The overarching goal of this study is to determine methods that can be successfully applied at a geologic CCS demonstration site for effective detection of

CO₂ and CH₄ leakage sources that may be associated with CCS. The guiding hypotheses of this study are:

1. Differences in greenhouse gas concentration data collected over weekends and during the week can be determined using statistical tools (such as Q-Q plots, Kruskal-Wallis H-Test, Kolmogorov-Smirnoff (K-S) Test, and Permutation tests).
2. The probable greenhouse gas emission sources can be determined using CO₂ and CH₄ concentration data coupled with wind speed and direction data through the construction of CBPFs over different time periods and over different ranges of concentrations

B.2 Data Description and Management

The Licor tower was set up on the University of Utah Ray Olpin Student Union building in June of 2016. Instruments have been continuously collecting CO₂, CH₄, H₂O, and wind data since that time. The Licor Eddy Covariance system collects data at a rate of ten hertz (Hz), or ten times per second, resulting in approximately 1.7 million rows of data per day. The raw data are currently housed on the CSER server maintained by the CSER group. Daily text files are combined into a text file using a bash script. The text file data are then uploaded into a MySQL database.

Time series were constructed of CO₂ and CH₄ concentration for each day of October to try and discern erroneous data that should be removed (Appendix C). Any coinciding high-concentration peaks between CO₂ and CH₄ remained in the dataset. Due to CO₂ and CH₄ being recorded using two different instruments on the Licor tower, it was assumed

that coinciding peaks were not erroneous. Extremely low CO₂ values (lower than 300 umol/mol) and extremely high CO₂ values (greater than 32,010 umol/mol) were removed. More data cleaning may be necessary to create a dataset that best represents concentrations that are occurring on the Union building roof.

B.3 Methods

Up to this point, the month of October 2016 was used for analysis despite having several months' worth of data. Descriptive statistics were calculated using RStudio. Boxplots were plotted for each day on an hourly basis to determine if there were any temporal trends. Furthermore, correlation tests were run to determine if certain variables were correlated with one another. At this time, statistical significance tests were not performed on the weekday or weekend data.

In addition to descriptive statistics, 99th percentile CO₂ and CH₄ CBPF were plotted using October's data, using the same methodology described in the Methods second of the preceding thesis. Plots were created using the entire dataset, using data collected during different time periods throughout the day, and using different concentration ranges. All scripts used to analyze the data and create CBPFs are located in Appendix D.

B.4 Ongoing Results

The cor() function in R was used to determine correlation between CO₂, CH₄, wind direction, wind speed, and hour the day using the spearman method. Table B.1 contains the correlation matrix for the entire October weekday dataset. Table B.2 shows the correlation matrix containing the highest correlation values, which took place at the wind

speed interval between 11 m/s and 11.99 m/s. From both correlation matrices, CO₂ and CH₄ concentration have the greatest correlation value.

Hourly boxplots were created using CO₂ and CH₄ concentration data as well as wind speed. These boxplots (Figures B.3-B.6) reflect the entire October weekday dataset on an hourly basis. Figure B.3 demonstrates that there are consistently high readings for CO₂ in the morning hours (5am-7am). The CH₄ boxplot (Figure B.5) shows high concentration values throughout the day. Both the CO₂ and CH₄ boxplots have numerous outliers, making it difficult to see any patterns in the means and quantiles of the data. Figure B.5 shows that the average wind speed increases in the early afternoon hours.

The 99th percentile CO₂ and CH₄ CBPFs were plotted using the entire October weekday dataset, along with the wind rose diagram (Figure B.6). The source of 99th percentile CO₂ concentration appears to be coming from the south-southeast and east directions whereas the 99th percentile CH₄ concentration appears to be coming from the west-southwest direction. A large kitchen exhaust outlet is located at 213° with respect to the tower, approximately 2 floors higher than the tower. A smaller exhaust outlet is located south-southeast of the tower on the same level of the Union roof that the tower is on.

To determine how the probable source location changes with respect to the tower location throughout the day, the 99th percentile concentration CBPFs were plotted at hourly intervals (Figures B.7-B.18). In the early hours from 4 am – 6 am, both CO₂ and CH₄ CBPFs illustrate a probable source located to the south east of the tower. The location of the emission sources varies throughout the day; however, the CH₄ source is

almost consistently originating in the southwest direction, which is the direction to the Union kitchen exhaust outlet.

Various high concentration ranges of CO₂ and CH₄ (95th-99th, 90th-95th, 75th-90th, and 50th-75th) were also plotted over the entire monthly weekday dataset (Figures B.19-B.22). The 95th-99th percentile as well as the 90th-95th percentile CO₂ and CH₄ CBPF all reveal an emission source location in the direction of the Union building kitchen exhaust outlet (Figures B.19-B.20). Emission source locations for lower ranges of concentrations begin to vary (Figures B.21-B.22).



Figure B.1 Union building roof experiment site. The blue circle indicates the Licor tower location. The red circle indicates the suspected GHG emission source.

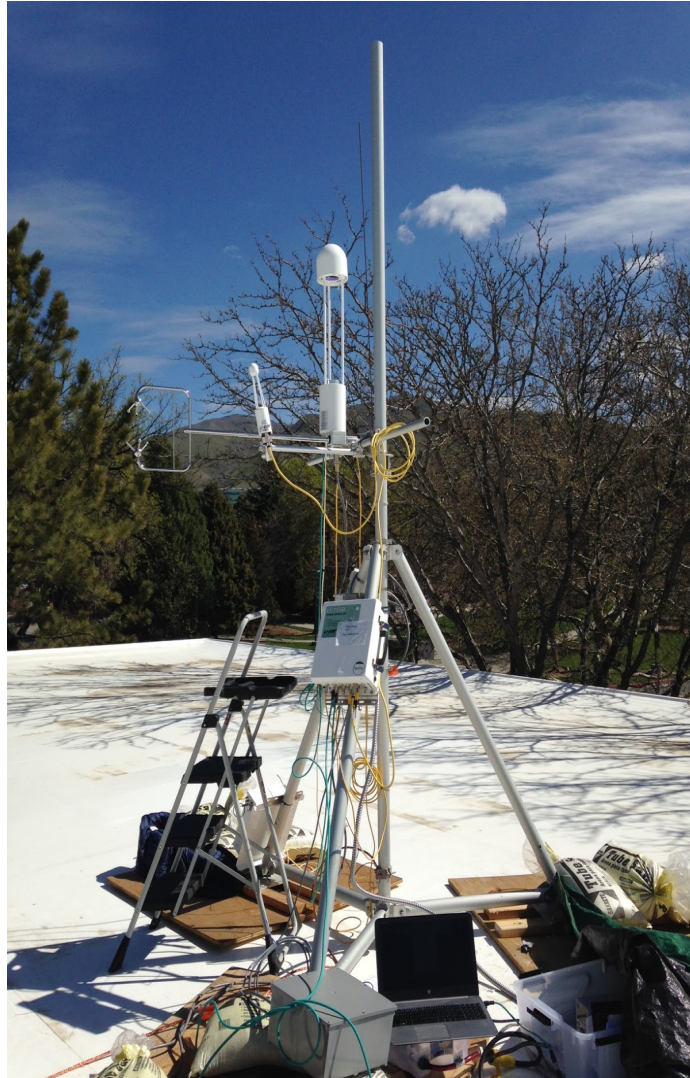
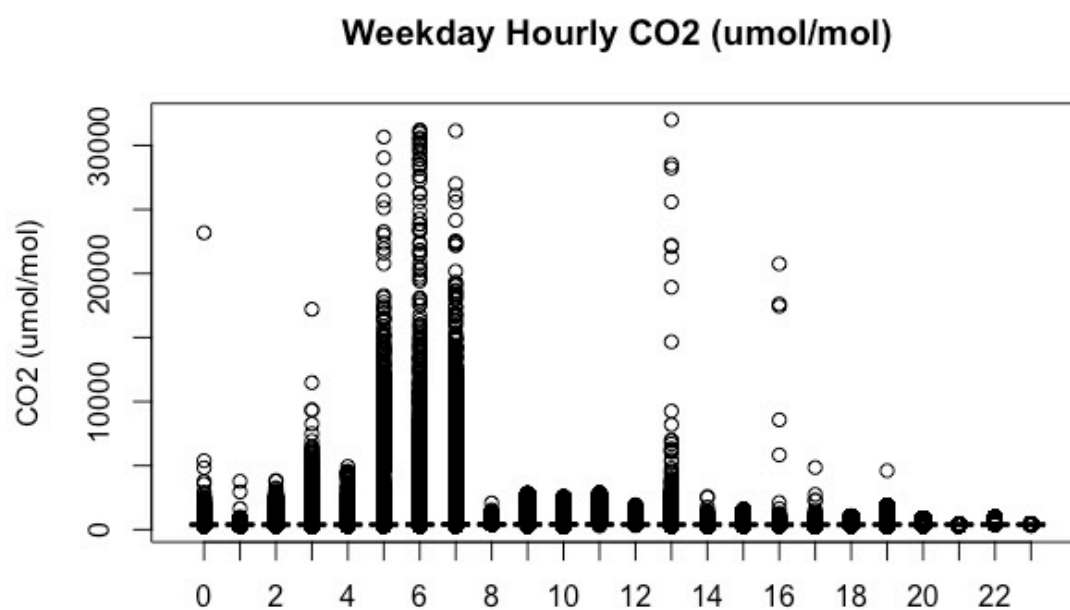
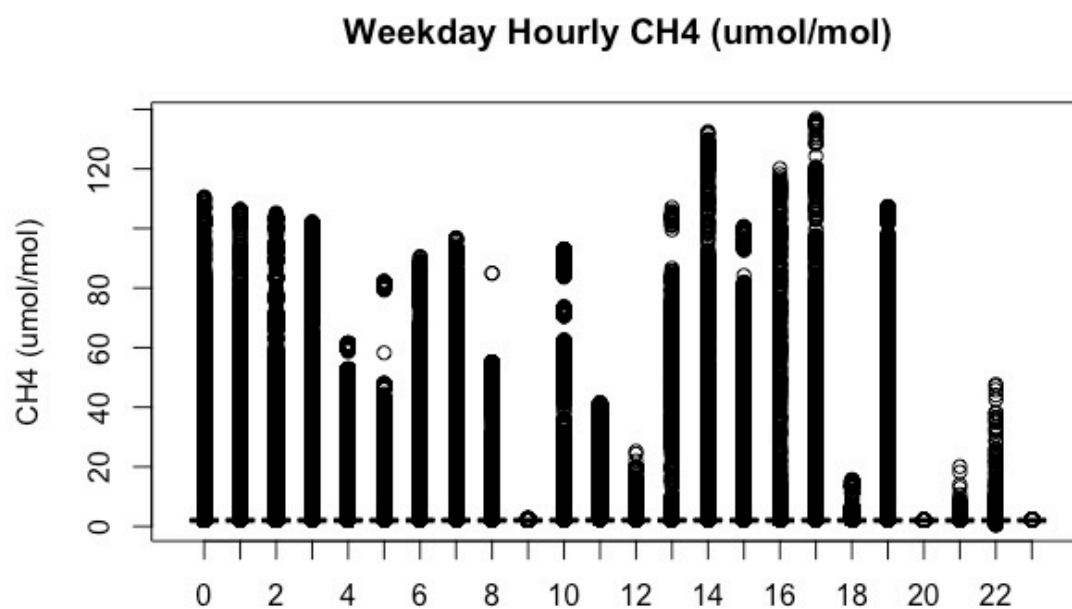


Figure B.2 Licor tower with instruments on the Union building roof

Figure B.3 Hourly CO₂ boxplot for October weekdaysFigure B.4 Hourly CH₄ boxplot for October weekdays

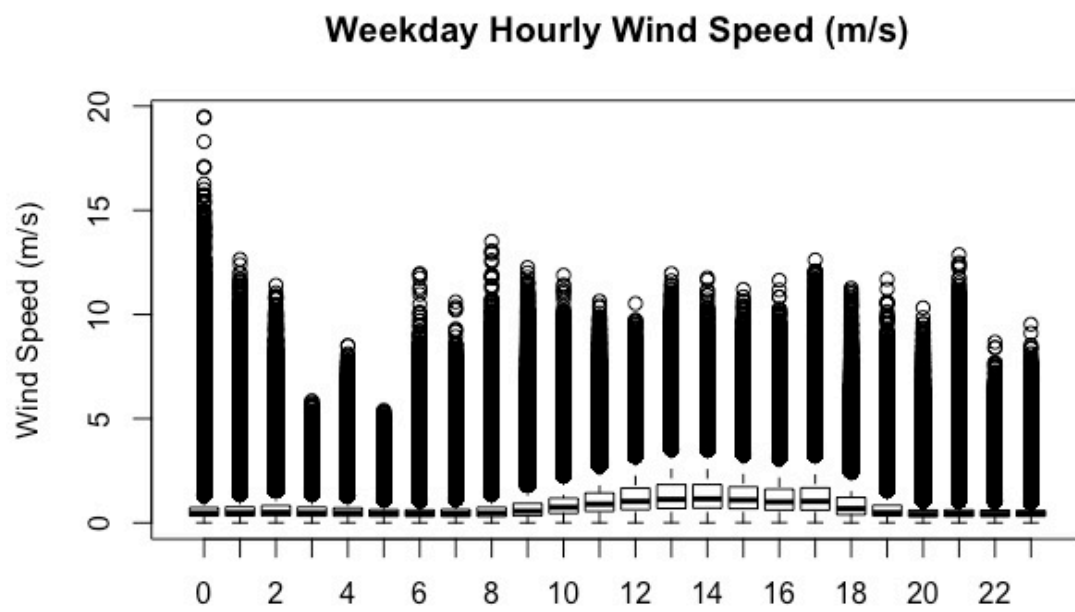


Figure B.5 Hourly wind speed boxplot for October weekdays

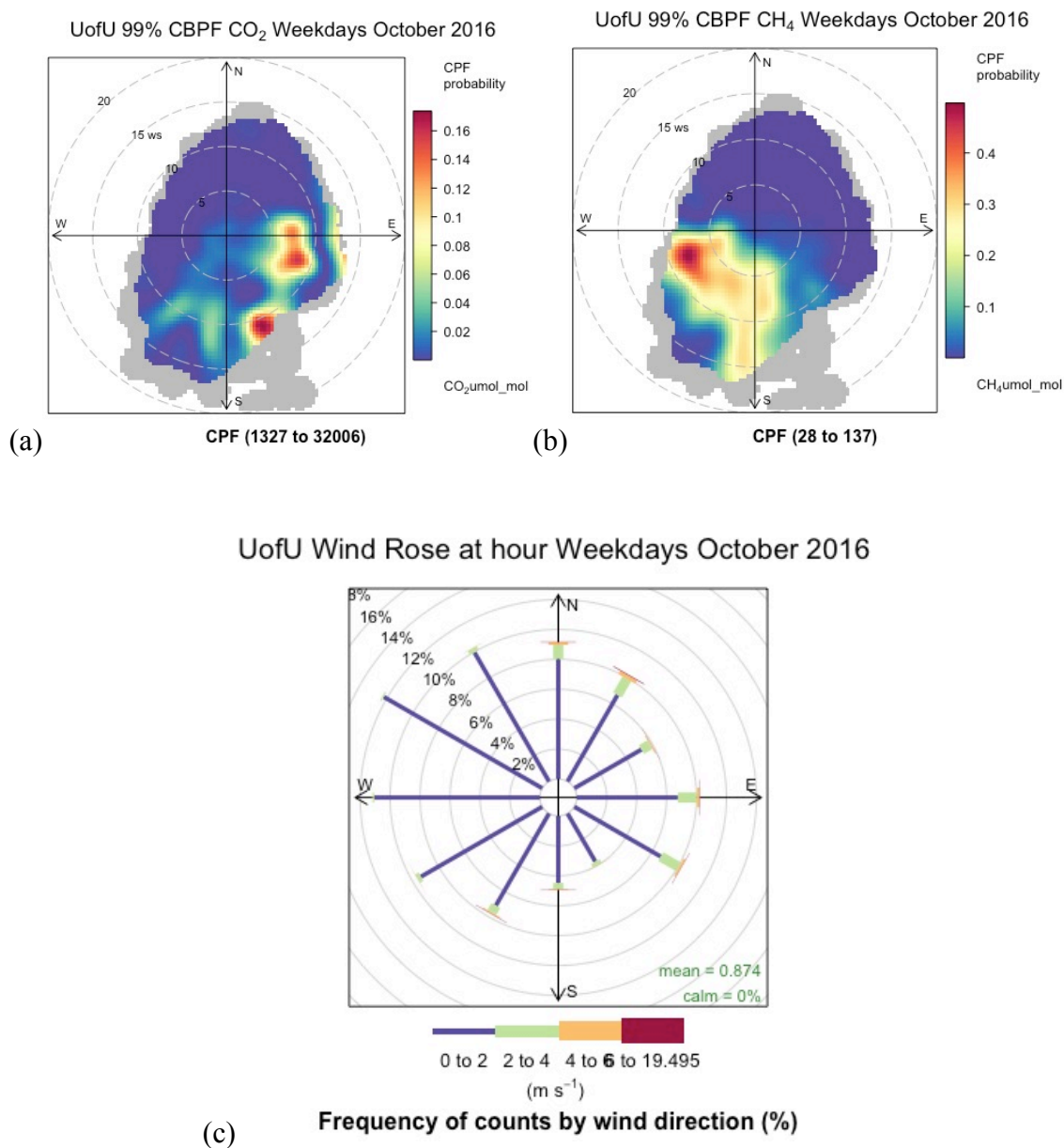


Figure B.6 (a) 99th percentile CO₂ CBPF, (b) 99th percentile CH₄ CBPF, (c) wind rose of entire weekday dataset.

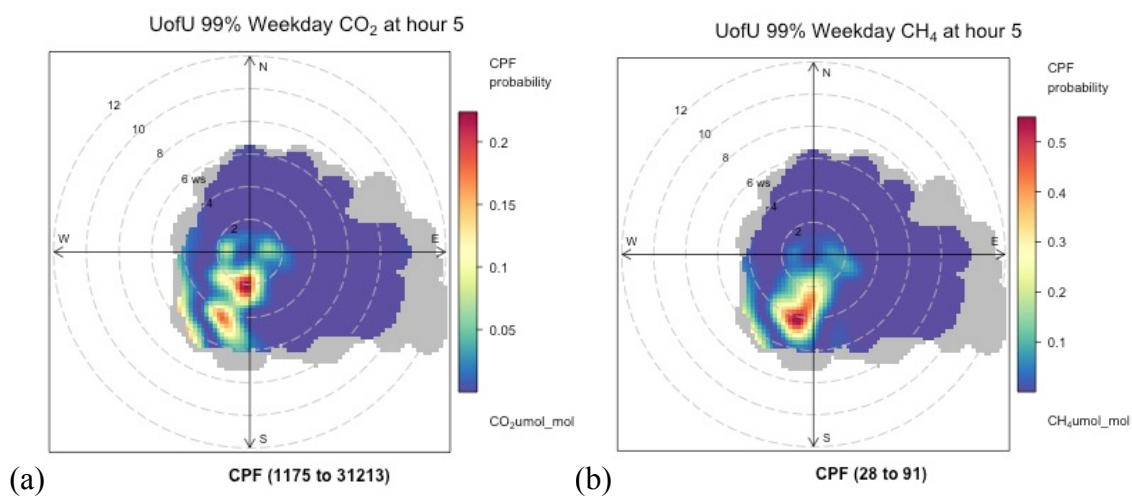


Figure B.7 (a) 99th percentile CO₂ CBPF, (b) 99th percentile CH₄ CBPF from 4 am to 5 am.

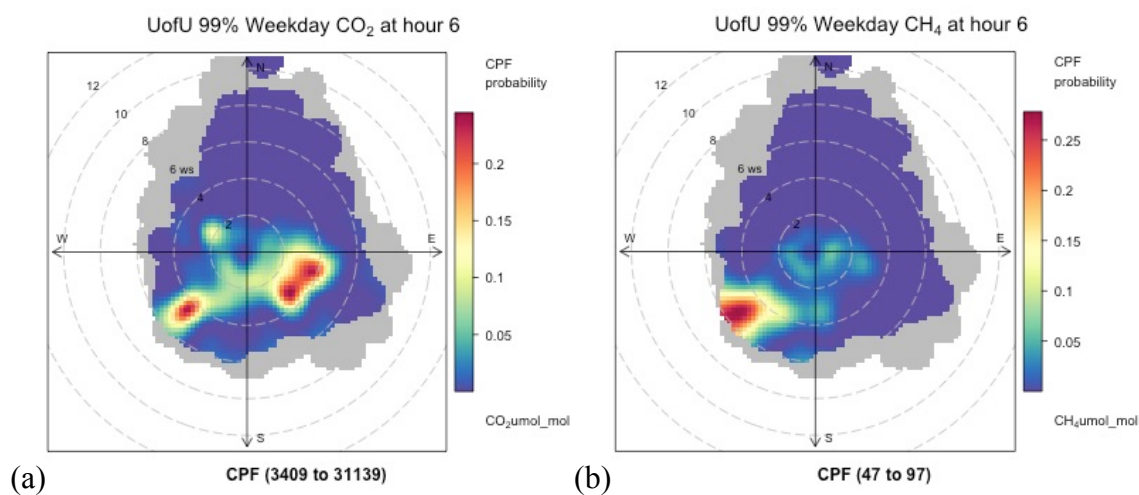


Figure B.8 (a) 99th percentile CO₂ CBPF, (b) 99th percentile CH₄ CBPF from 5 am to 6 am.

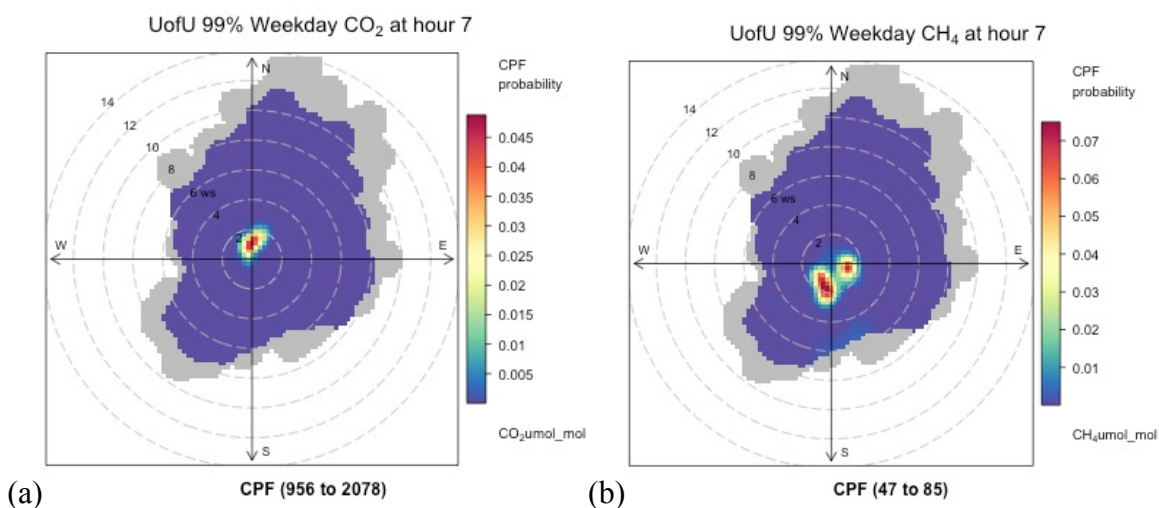


Figure B.9 (a) 99th percentile CO₂ CBPF, (b) 99th percentile CH₄ CBPF from 6 am to 7 am.

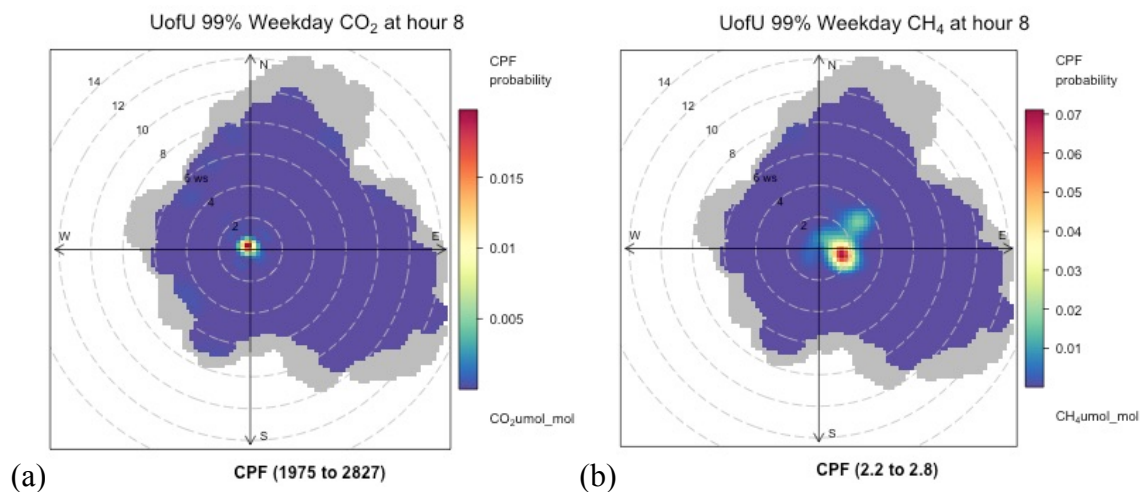


Figure B.10 (a) 99th percentile CO₂ CBPF, (b) 99th percentile CH₄ CBPF from 7 am to 8 am.

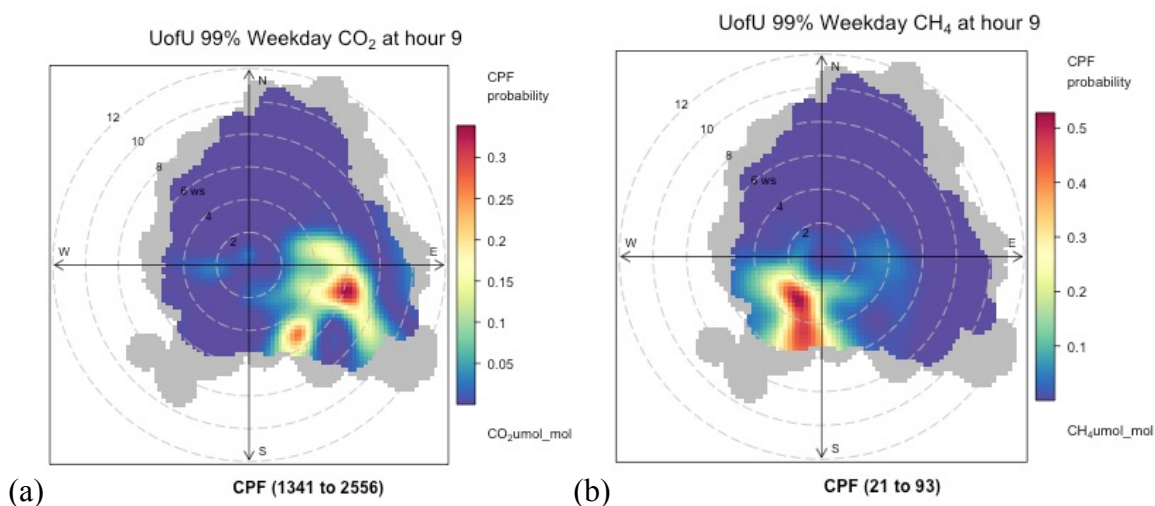


Figure B.11 (a) 99th percentile CO₂ CBPF, (b) 99th percentile CH₄ CBPF from 8 am to 9 am.

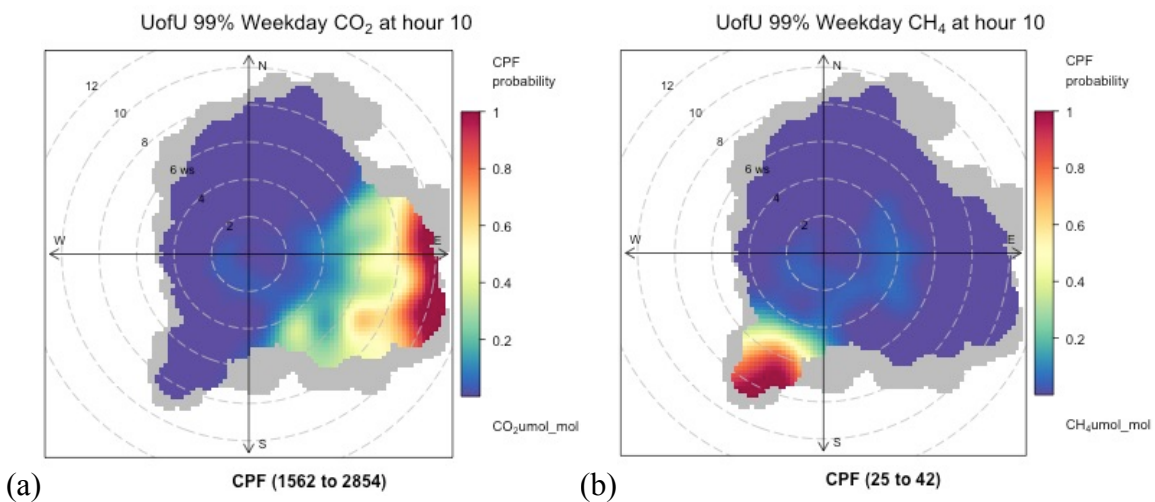


Figure B.12 (a) 99th percentile CO₂ CBPF, (b) 99th percentile CH₄ CBPF from 9 am to 10 am.

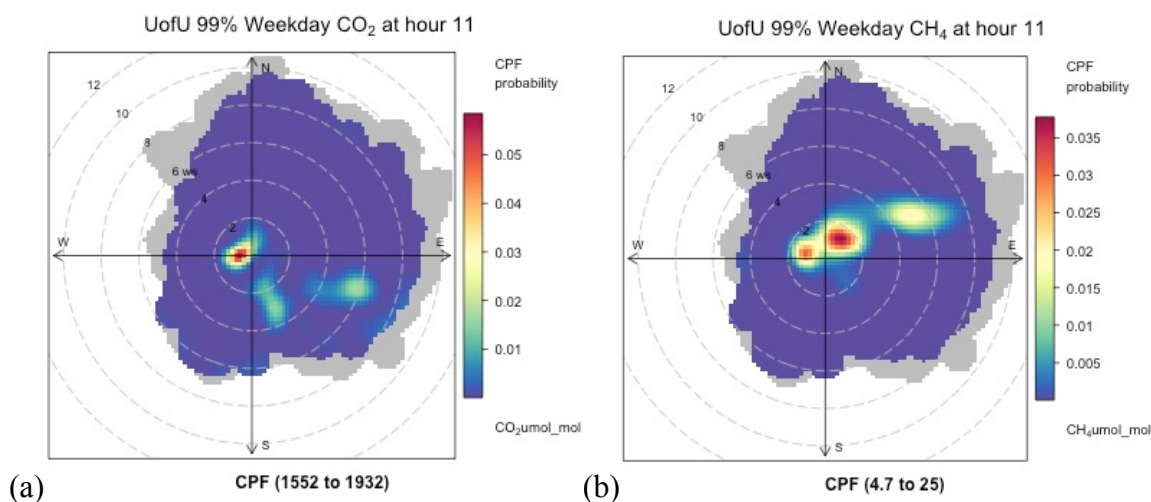


Figure B.13 (a) 99th percentile CO₂ CBPF, (b) 99th percentile CH₄ CBPF from 10 am to 11 am.

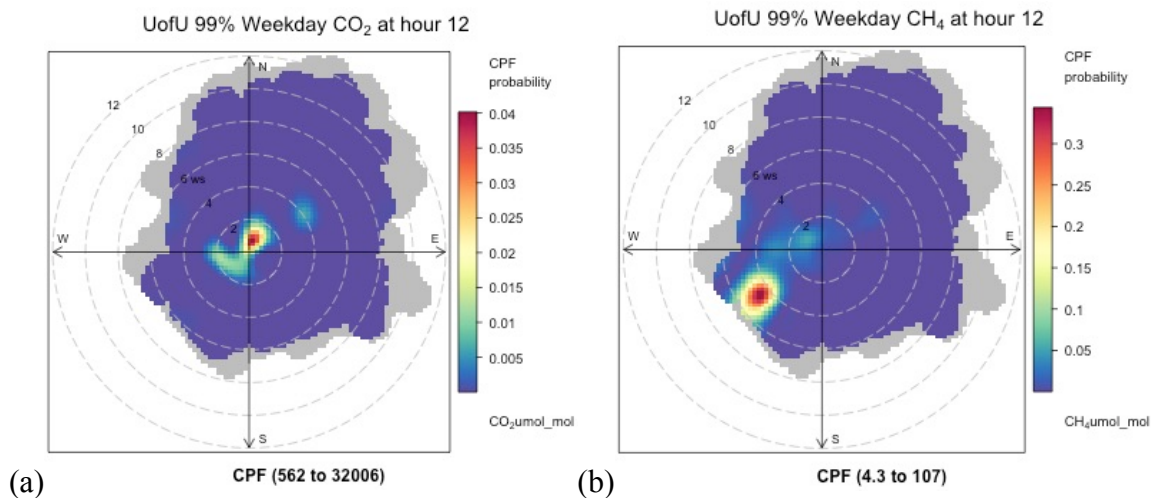


Figure B.14 (a) 99th percentile CO₂ CBPF, (b) 99th percentile CH₄ CBPF from 11 am to 12 pm.

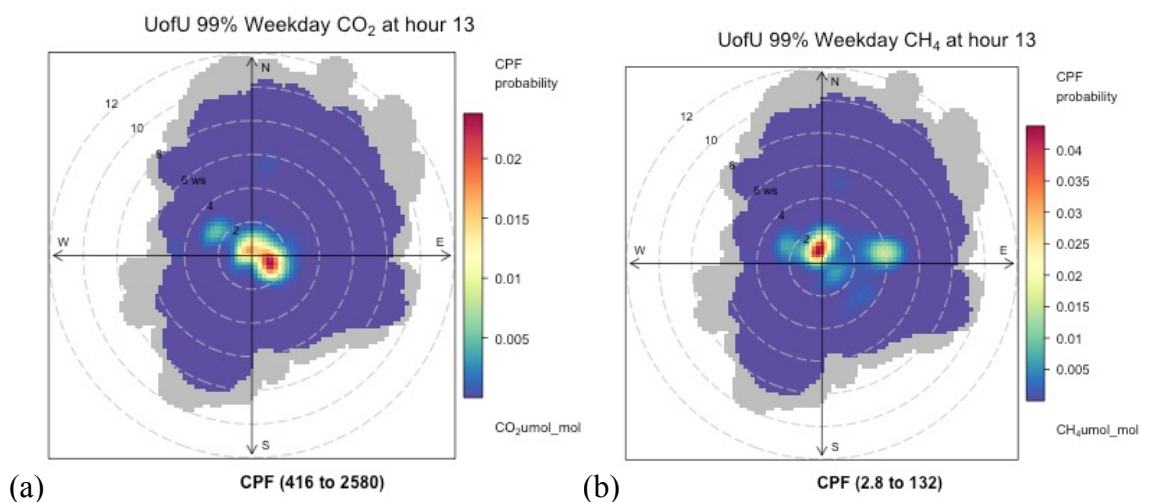


Figure B.15 (a) 99th percentile CO₂ CBPF, (b) 99th percentile CH₄ CBPF from 12 pm to 1 pm.

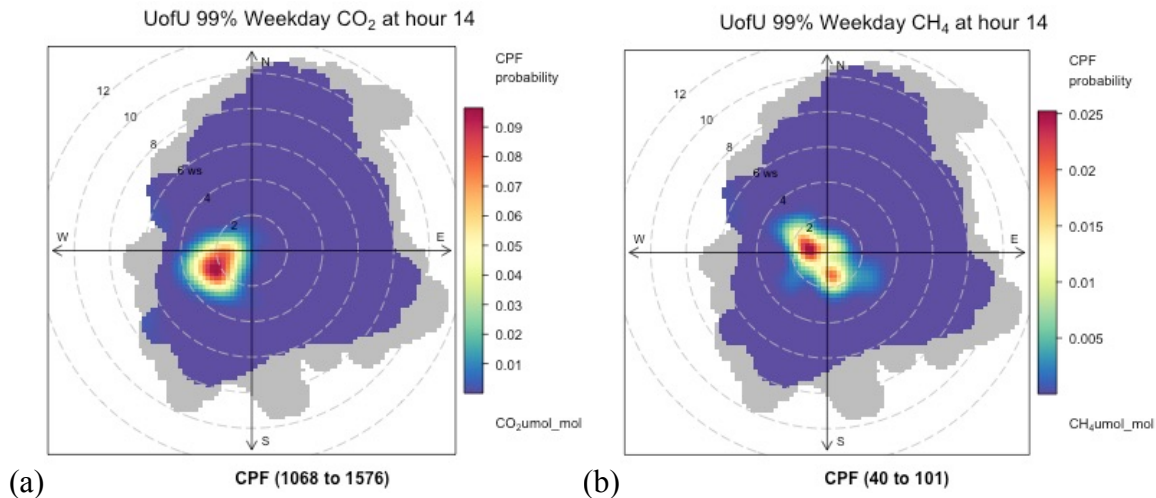


Figure B.16 (a) 99th percentile CO₂ CBPF, (b) 99th percentile CH₄ CBPF from 1 pm to 2 pm.

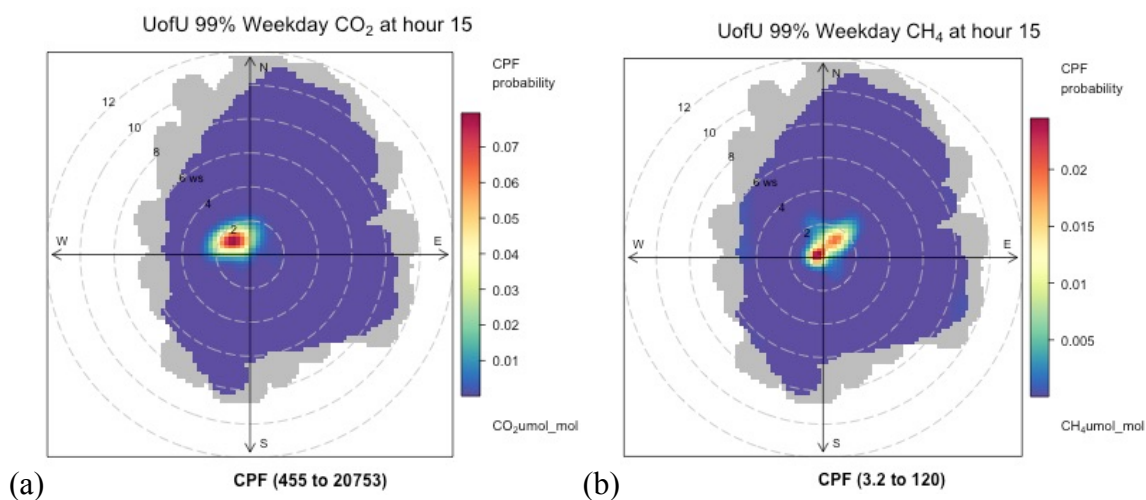


Figure B.17 (a) 99th percentile CO₂ CBPF, (b) 99th percentile CH₄ CBPF from 2 pm to 3 pm.

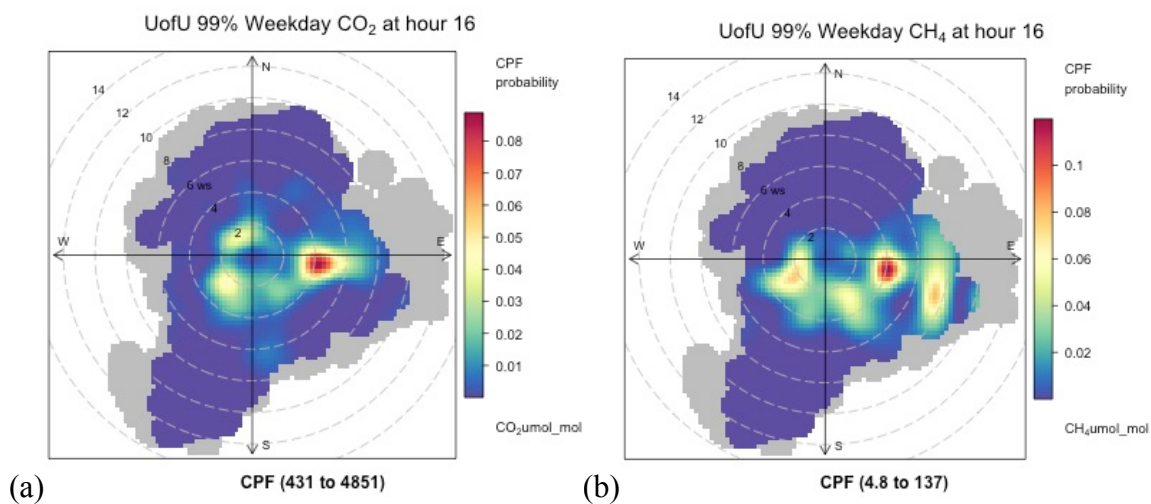


Figure B.18 (a) 99th percentile CO₂ CBPF, (b) 99th percentile CH₄ CBPF from 3 pm to 4 pm.

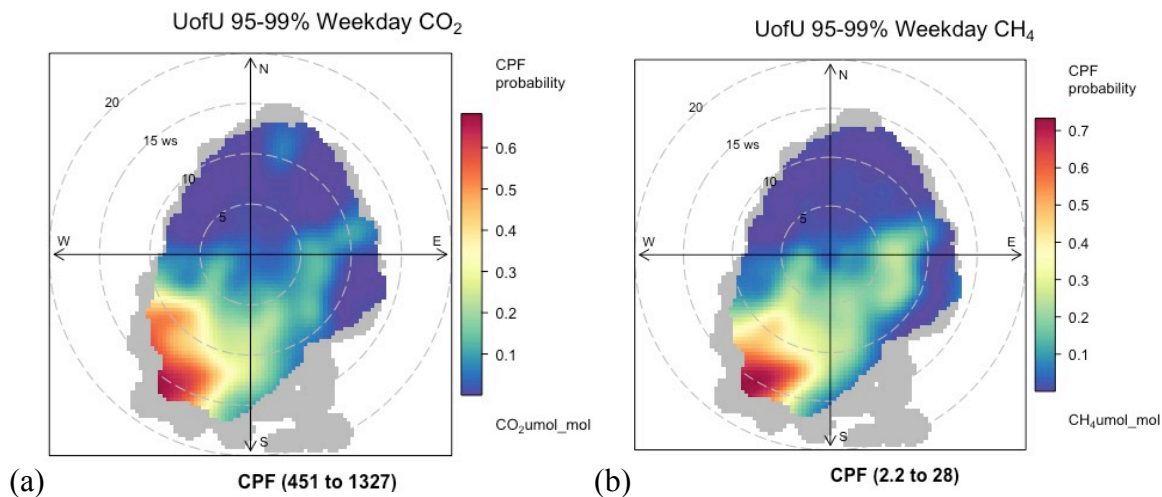


Figure B.19 (a) 95th-99th percentile CO₂ CBPF, (b) 95th-99th percentile CH₄ CBPF.

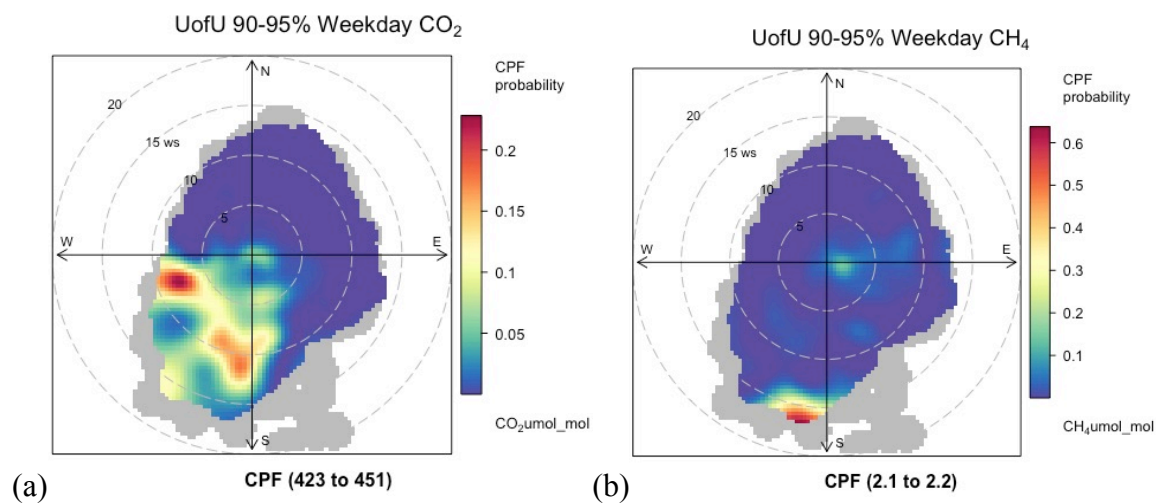


Figure B.20 (a) 90th-95th percentile CO₂ CBPF, (b) 90th-95th percentile CH₄ CBPF.

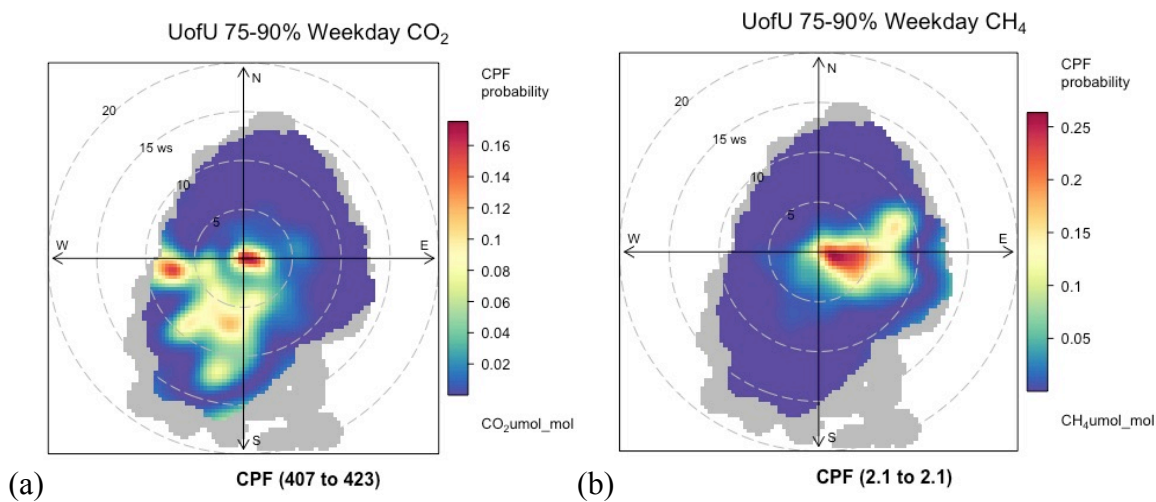


Figure B.21 (a) 75th-90th percentile CO₂ CBPF, (b) 75th-90th percentile CH₄ CBPF.

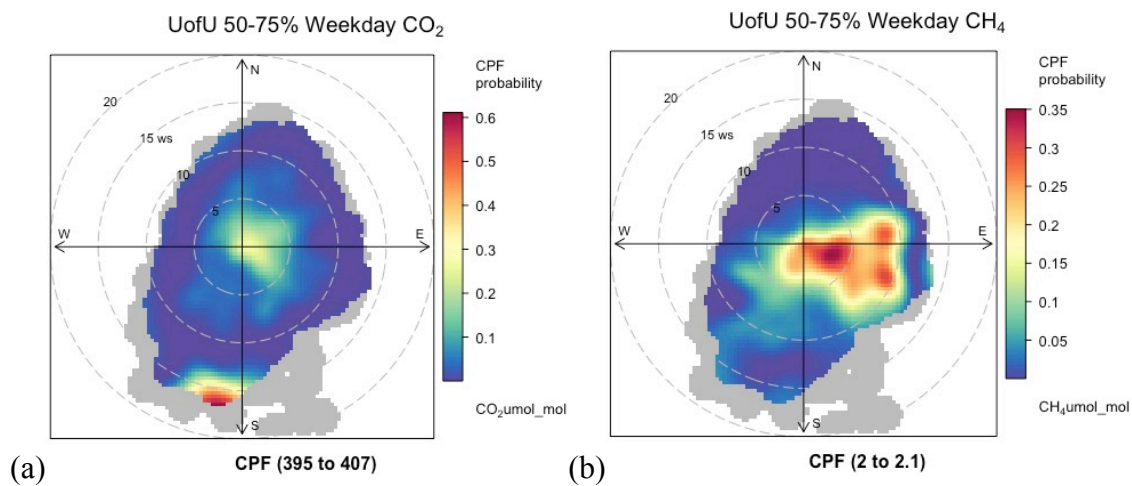


Figure B.22 (a) 50th-75th percentile CO₂ CBPF, (b) 50th-75th percentile CH₄ CBPF.

Table B.1 Correlation matrix using the spearman method for entire weekday dataset

	CO2(umol/mol)	CH4(umol/mol)	Wind Dir.	Wind Speed	Hour
CO2(umol/mol)		0.32551	0.06463	-0.17101	-0.00005
CH4(umol/mol)	0.32551		-0.05801	-0.01459	-0.00459
Wind Dir.	0.06463	-0.05801		-0.34260	-0.03395
Wind Speed	-0.17101	-0.01459	-0.34260		0.06392
Hour	-0.00005	-0.00459	-0.03395	0.06392	

Table B.2 Correlation matrix using the spearman method for weekday dataset with winds speeds between 10 m/s and 11 m/s

	CO2(umol/mol)	CH4(umol/mol)	Wind Dir.	Wind Speed	Hour
CO2(umol/mol)		0.76097	0.40833	0.16802	-0.73672
CH4(umol/mol)	0.76097		0.47697	0.16691	-0.64467
Wind Dir.	0.40833	0.47697		0.00437	-0.35334
Wind Speed	0.16802	0.16691	0.00437		-0.10811
Hour	-0.73672	-0.64467	-0.35334	-0.10811	

APPENDIX C

TIME SERIES PLOTS CREATED FOR DATA CLEANING PURPOSES

Below are time series plots for CO₂ and CH₄ concentration for every weekday in October (Figures C.1-C.21). The data used to make the time series have not been cleaned.

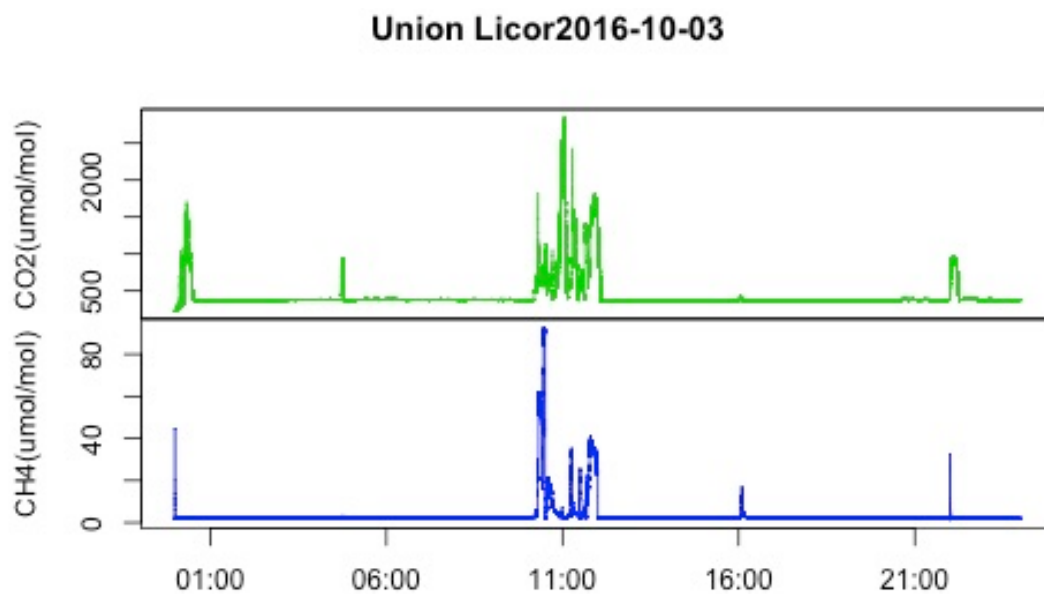


Figure C.1 CO₂ and CH₄ time series for 10/3/2016

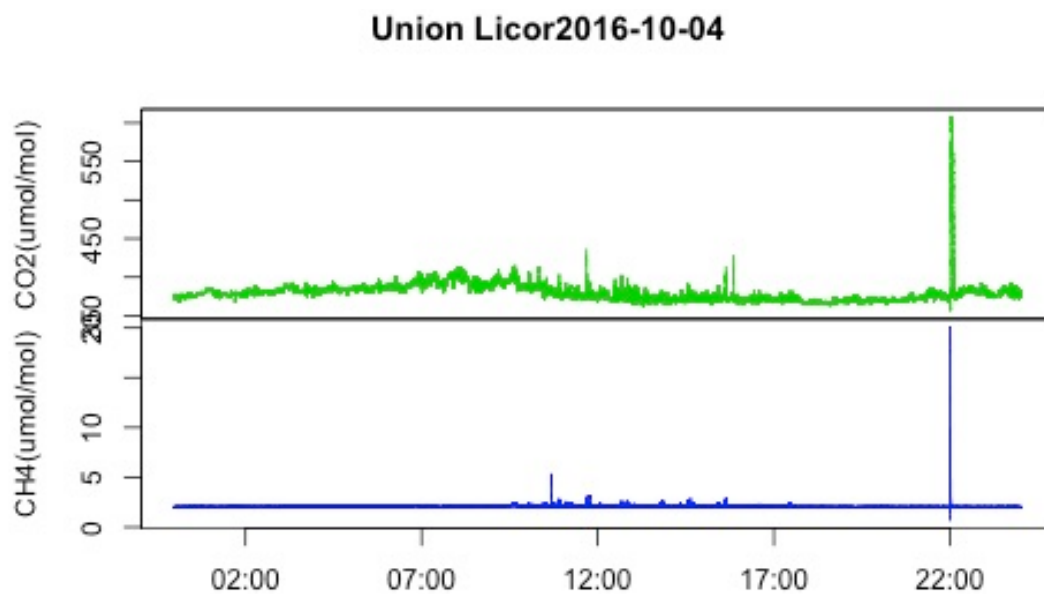


Figure C.2 CO₂ and CH₄ time series for 10/4/2016

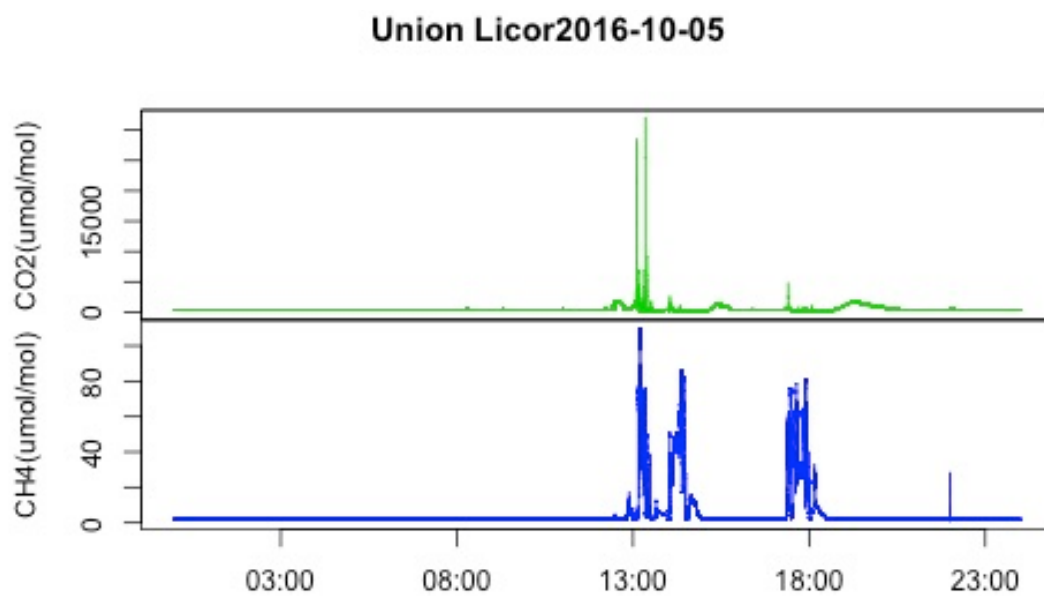


Figure C.3 CO₂ and CH₄ time series for 10/5/2016

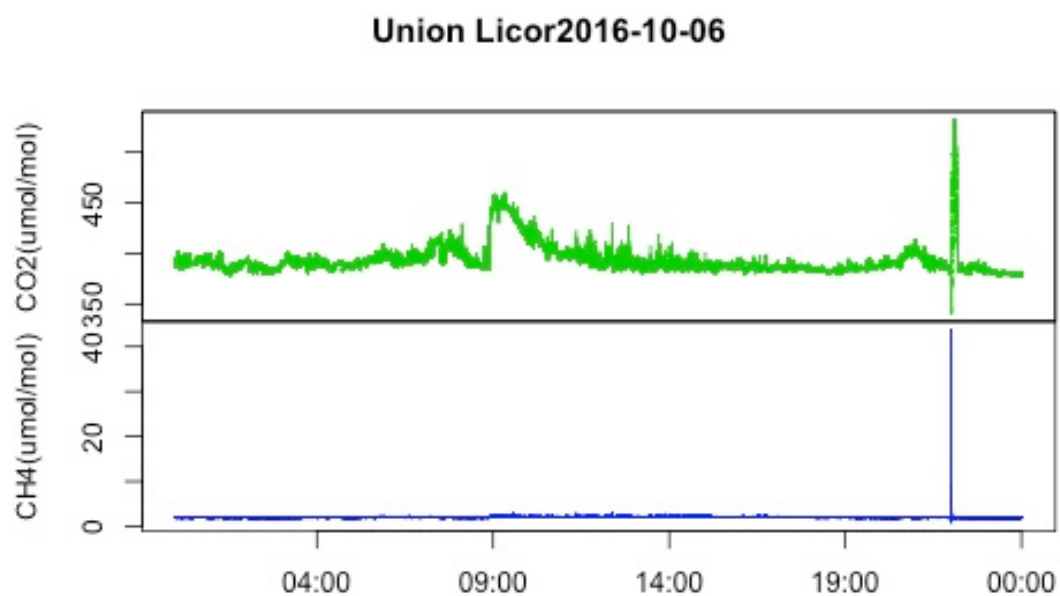


Figure C.4 CO₂ and CH₄ time series for 10/6/2016

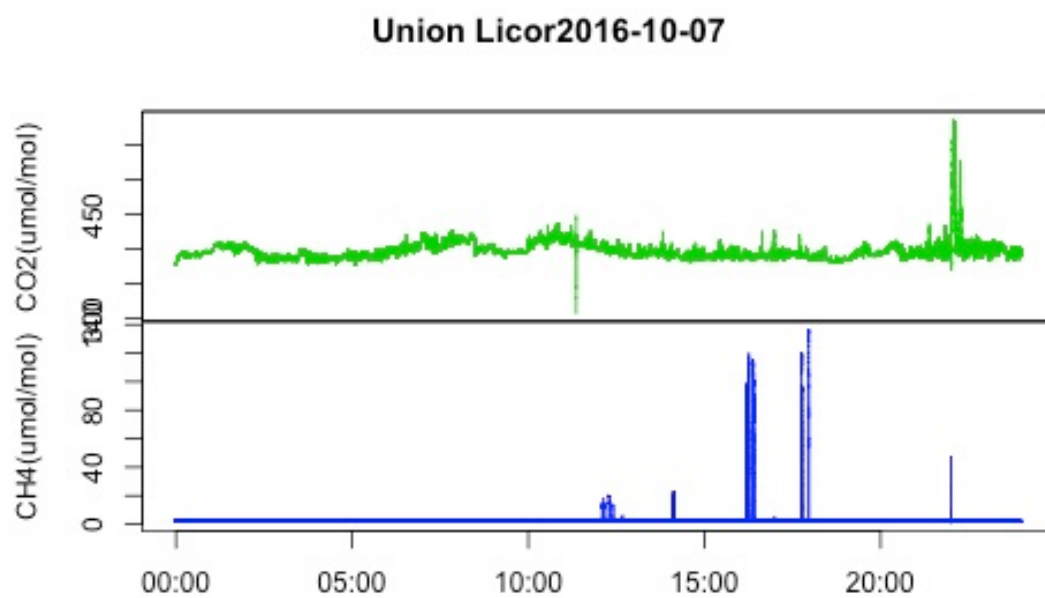


Figure C.5 CO₂ and CH₄ time series for 10/7/2016

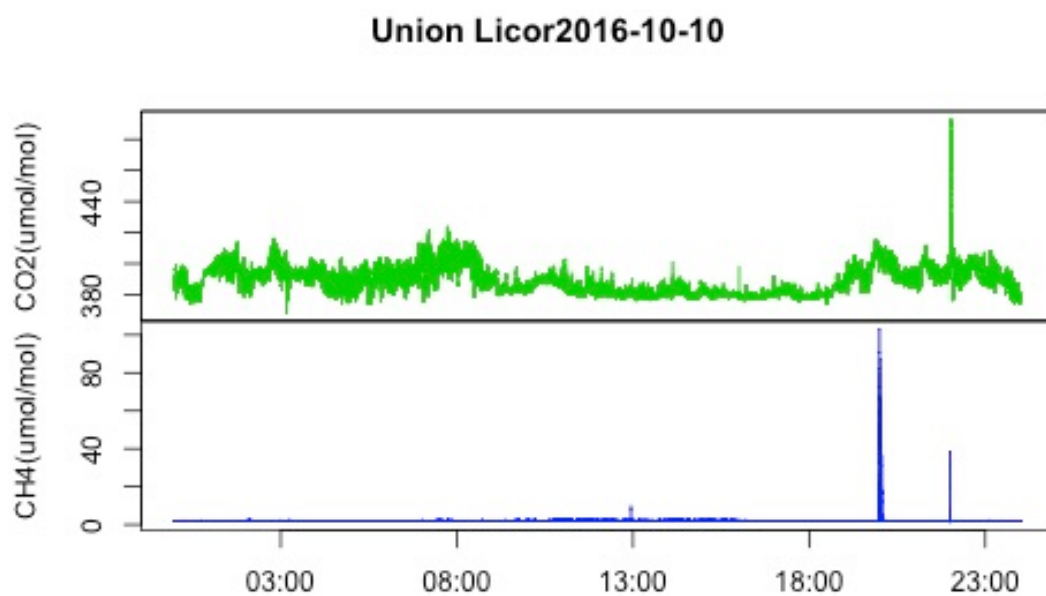


Figure C.6 CO₂ and CH₄ time series for 10/10/2016

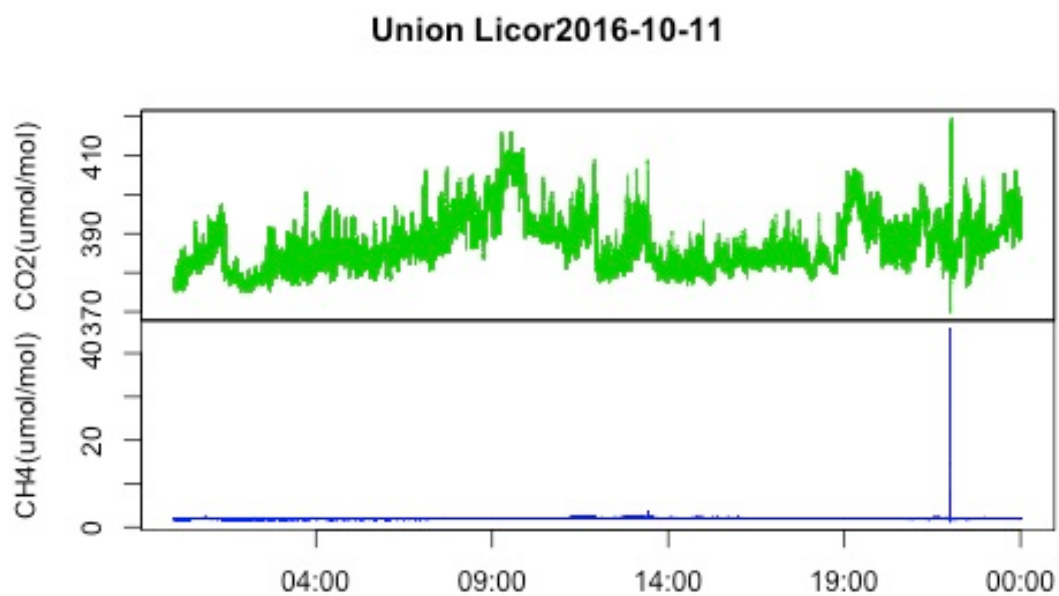


Figure C.7 CO₂ and CH₄ time series for 10/11/2016

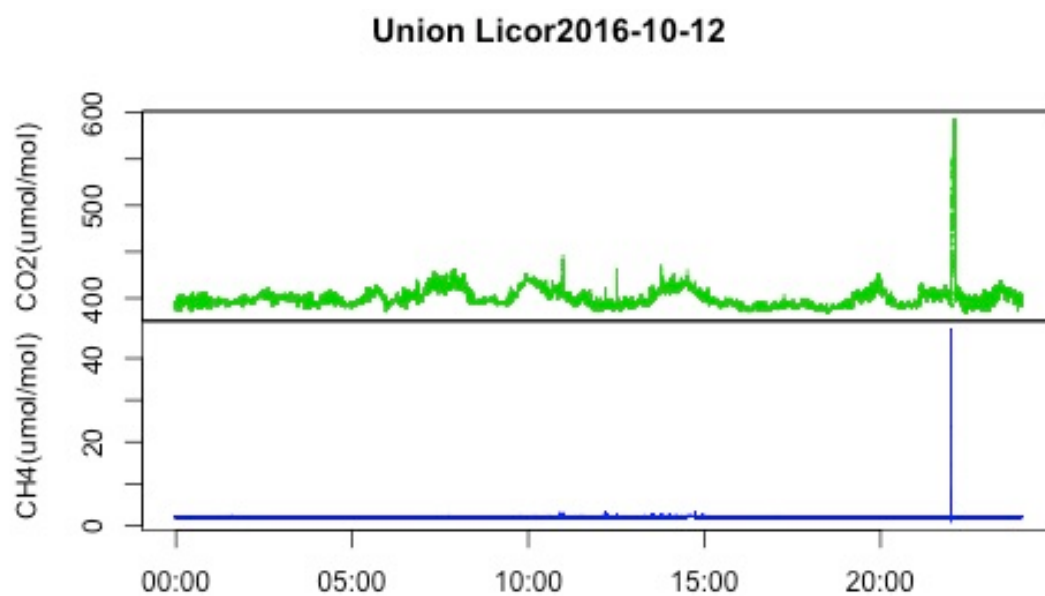


Figure C.8 CO₂ and CH₄ time series for 10/12/2016

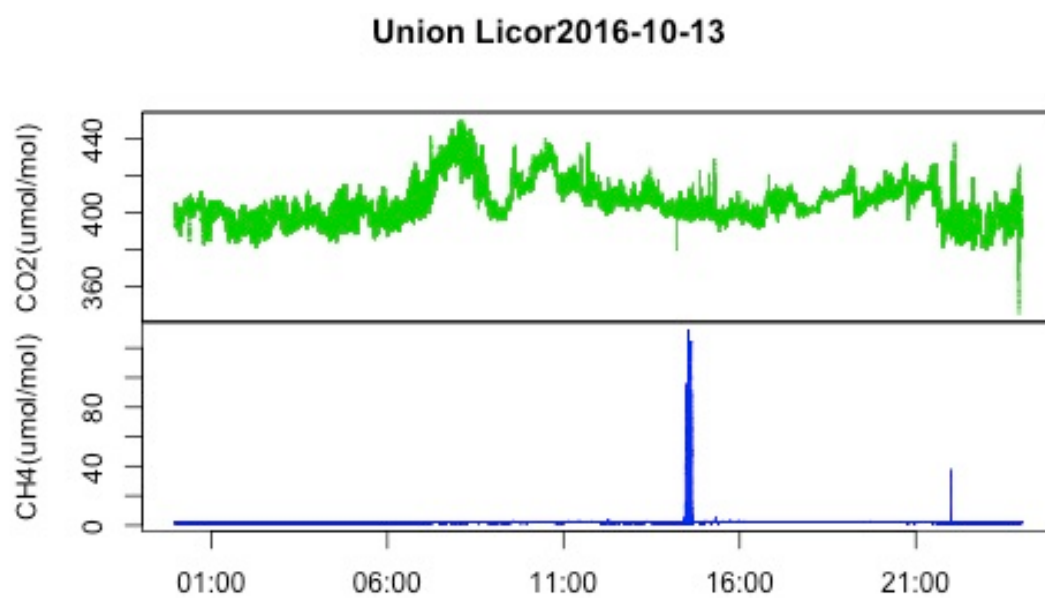


Figure C.9 CO₂ and CH₄ time series for 10/13/2016

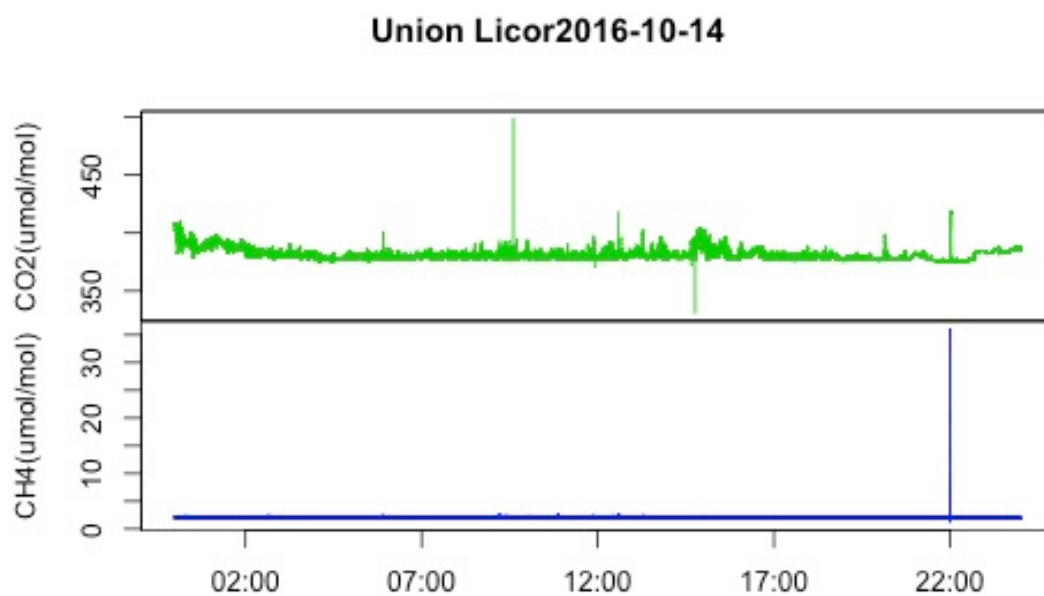


Figure C.10 CO₂ and CH₄ time series for 10/14/2016

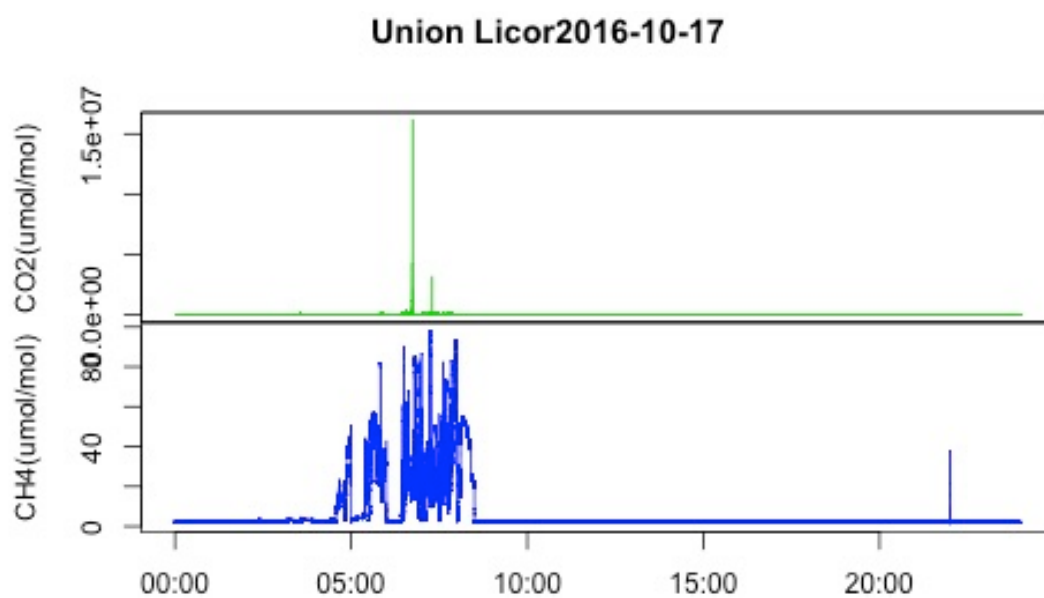


Figure C.11 CO₂ and CH₄ time series for 10/17/2016

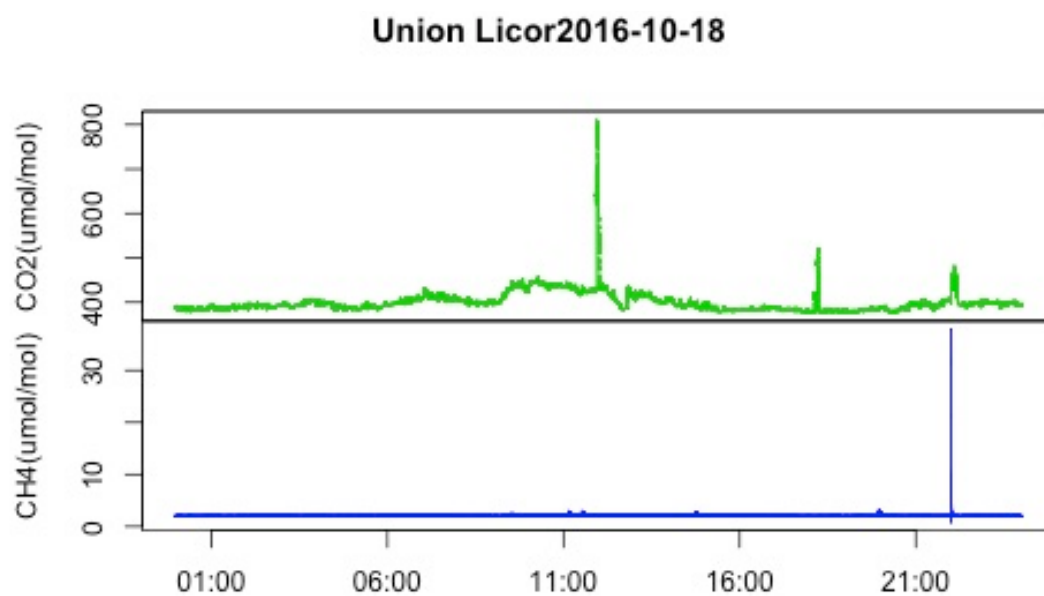


Figure C.12 CO₂ and CH₄ time series for 10/18/2016

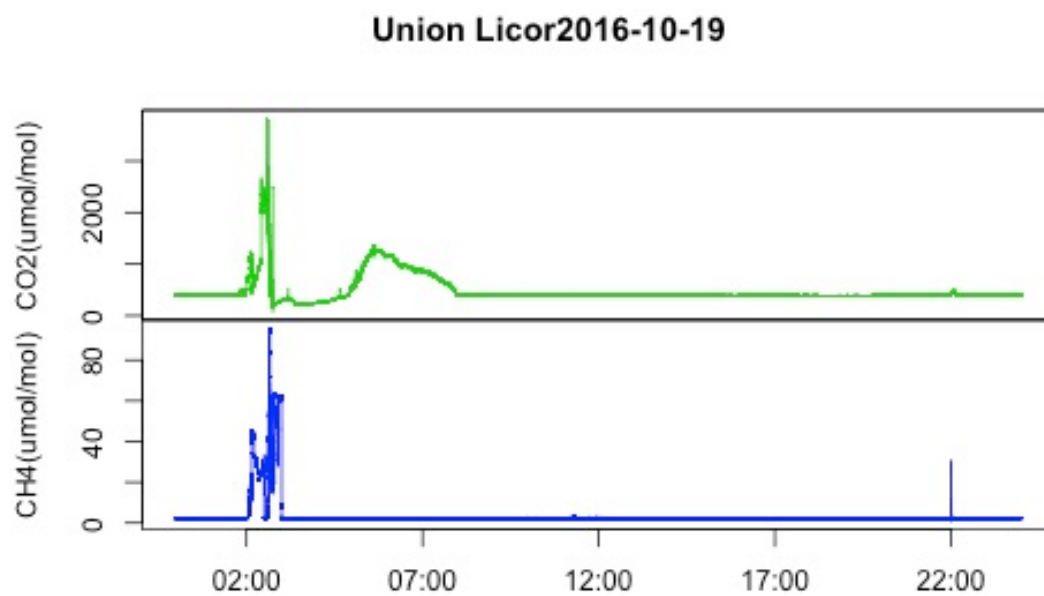


Figure C.13 CO₂ and CH₄ time series for 10/19/2016

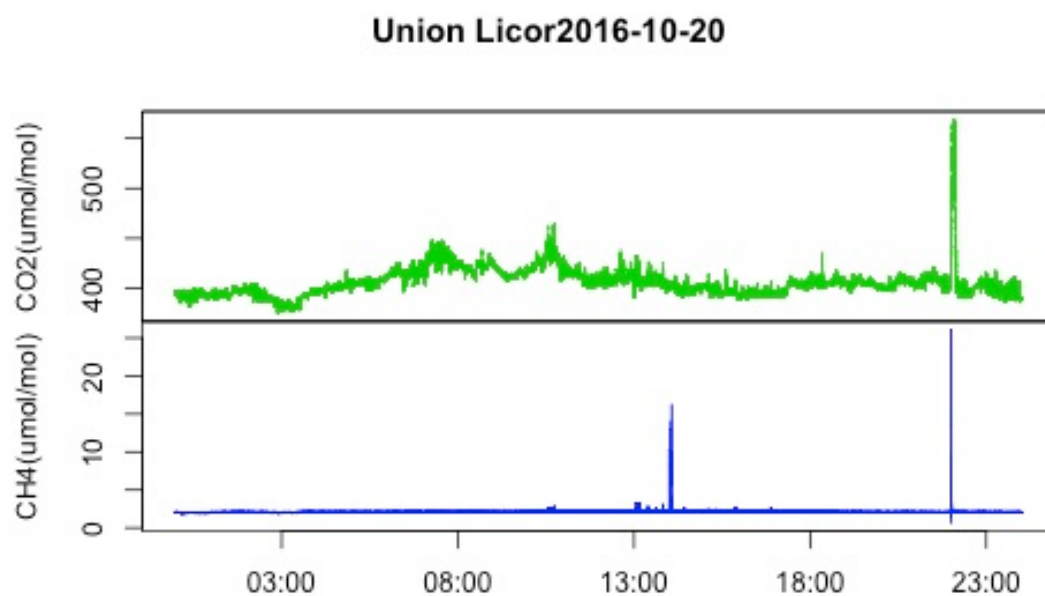


Figure C.14 CO₂ and CH₄ time series for 10/20/2016

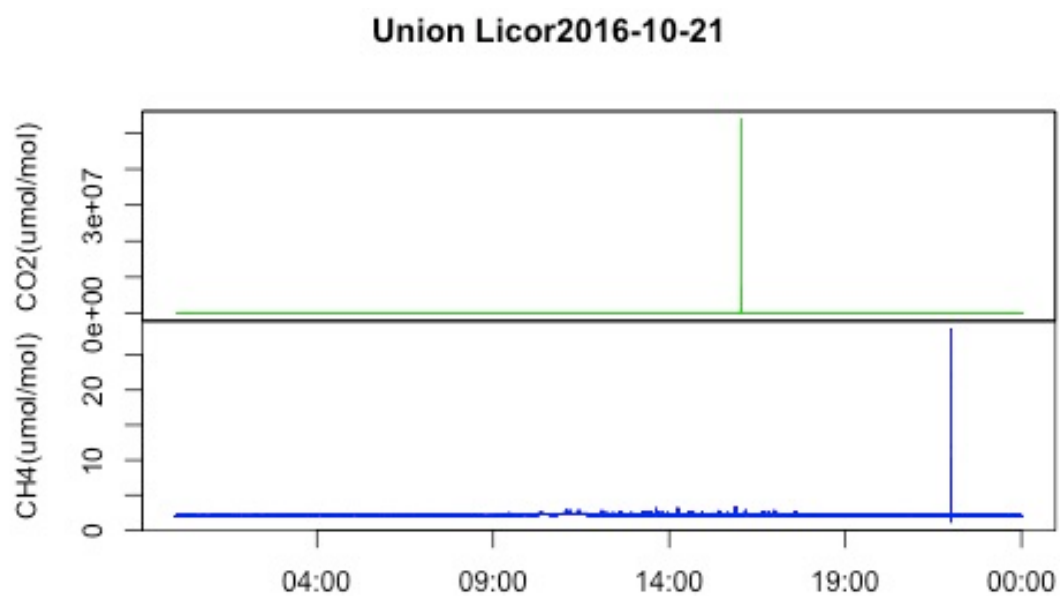
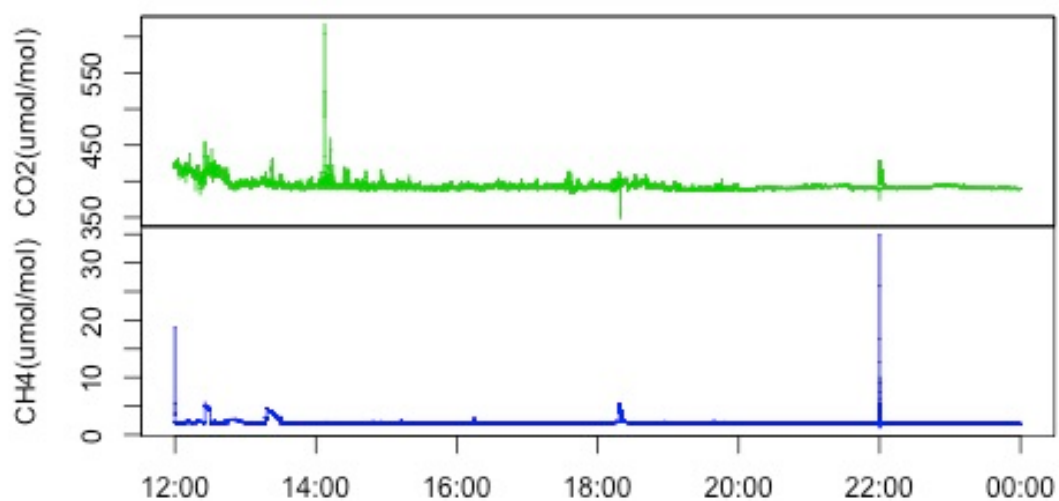
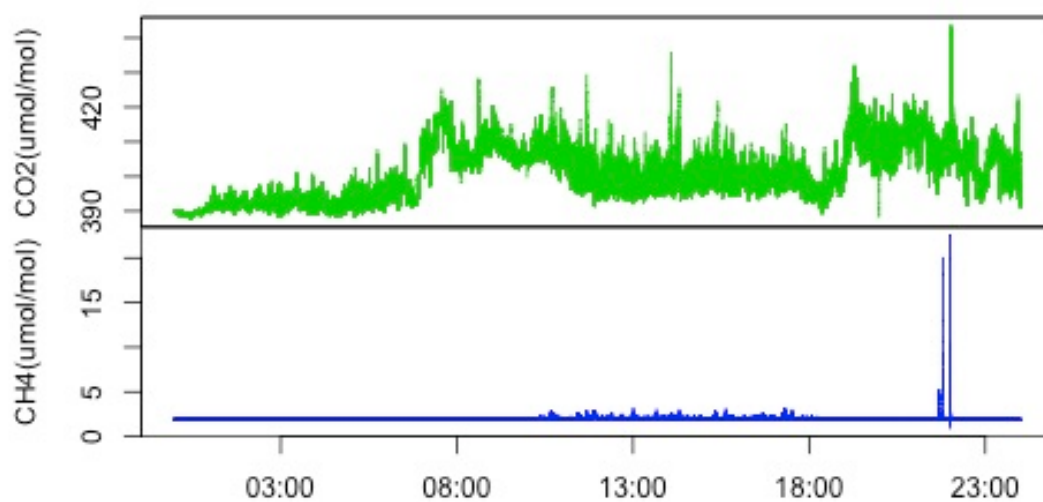


Figure C.15 CO₂ and CH₄ time series for 10/21/2016

Union Licor2016-10-24Figure C.16 CO₂ and CH₄ time series for 10/24/2016**Union Licor2016-10-25**Figure C.17 CO₂ and CH₄ time series for 10/25/2016

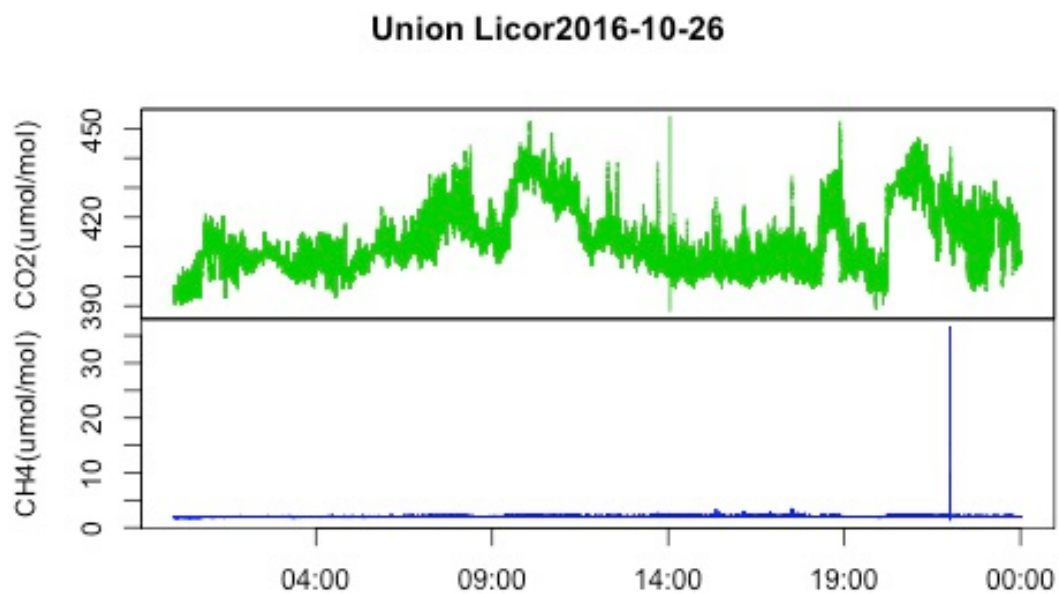


Figure C.18 CO₂ and CH₄ time series for 10/26/2016

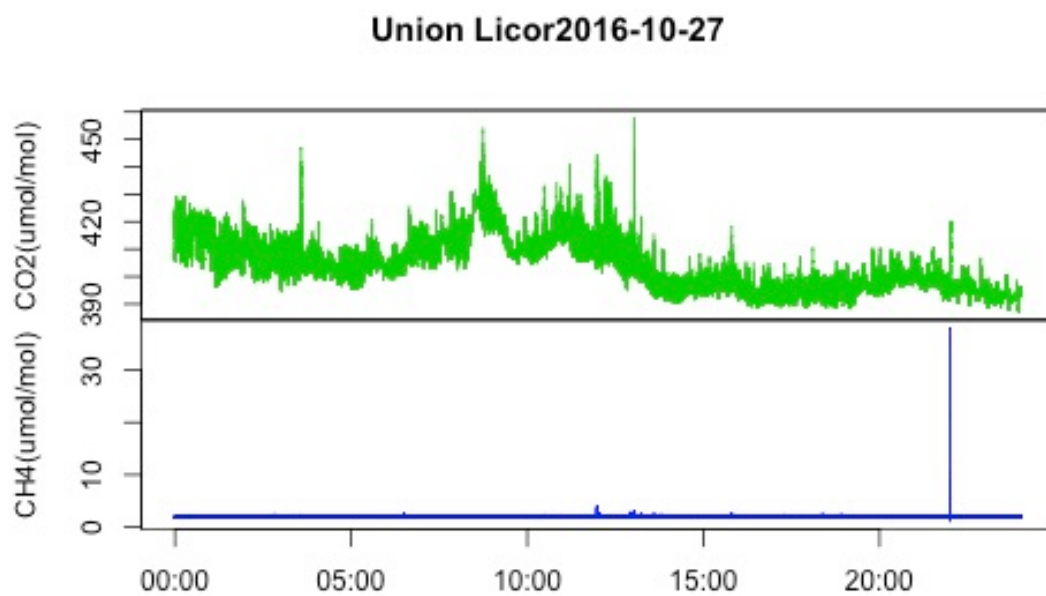
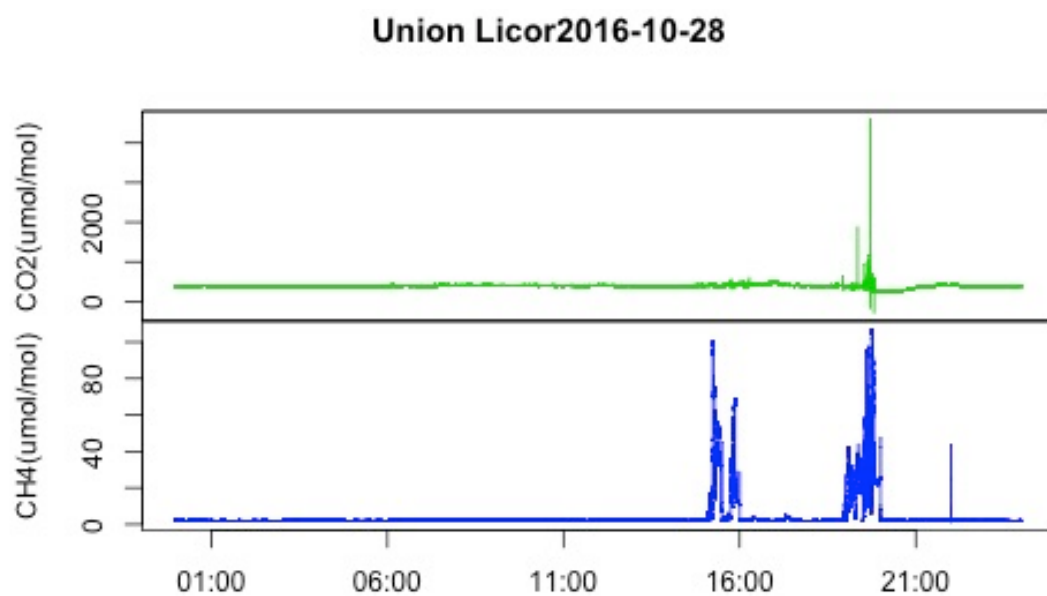
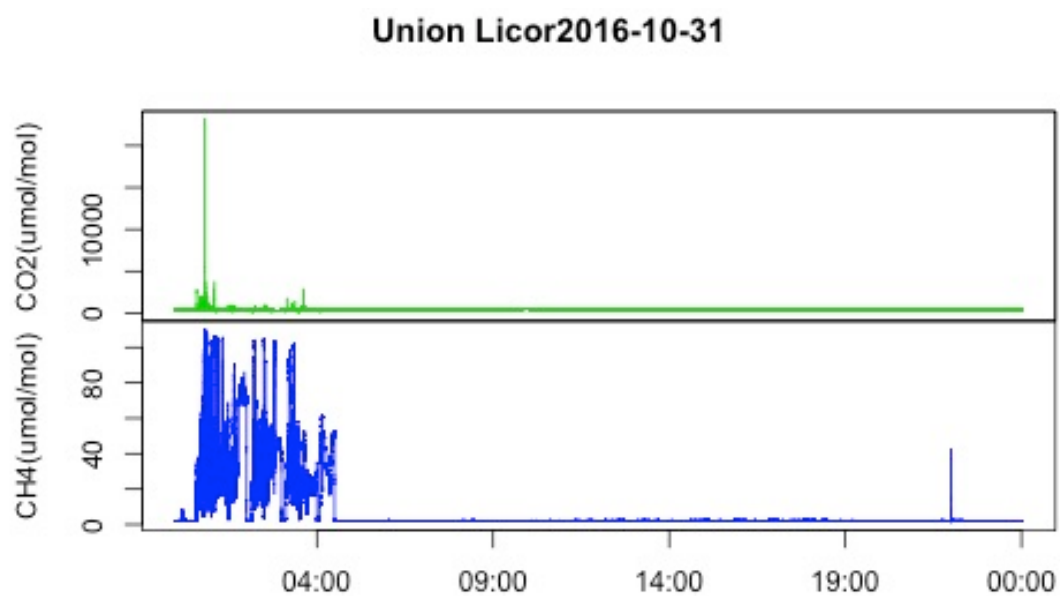


Figure C.19 CO₂ and CH₄ time series for 10/27/2016

Figure C.20 CO₂ and CH₄ time series for 10/28/2016Figure C.21 CO₂ and CH₄ time series for 10/31/2016

APPENDIX D

R SCRIPTS FOR DATA ANALYSIS ON UNIVERSITY OF UTAH CAMPUS UNION BUILDING FOR OCTOBER 2016

This scripts downloads raw data from the MySQL server for the month of October, 2016. This script creates first daily data, then monthly weekday data. The script runs through correlation tests, boxplot plotting, and CBPF plotting.

```
setwd("~/Dropbox (Personal)/Research/_ANALYSIS/UofU/Union/Oct")

con<-dbConnect(RMySQL::MySQL(), username ="mwash", password="qXfjc01B$C0o",
host="155.98.6.253", dbname="LiCor_EddyFlux")
dbListFields(con, 'Union10012016_10232016') ##List Fields

rs = dbSendQuery(con, "SELECT DISTINCT Date FROM Union10012016_10232016 ORDER
BY Date")
OctDates<-fetch(rs, n=-1)
##Only want weekdays
OctDates2<-OctDates[c(3,4,5,6,7,10,11,12,13,14,17,18,19,20,21),]
OctDates2<-as.data.frame(OctDates2)
rm(OctDates)
datasets <- matrix(ncol=ncol(OctDates2), nrow=(nrow(OctDates2)))
for(i in 1:(nrow(OctDates2))){
  DBDate<-OctDates2[i,]
  sqlStatement <- paste("SELECT row_names, Date, Time,CO2umol_mol,
CH4umol_mol, wd, ws FROM Union10012016_10232016 WHERE Date='",DBDate, "'",
sep="")
  rs = dbSendQuery(con, sqlStatement)
  title<-gsub("-", "",paste("data", DBDate, sep = ''))
  datasets[i,]<-title
  assign(title, fetch(rs, n=-1))
  dbClearResult(rs)
}
```



```

datasets2<-as.list(data.frame(datasets))

save(data20161016,file="data20161016.Rda")
save(data20161017,file="data20161017.Rda")
save(data20161018,file="data20161018.Rda")
save(data20161019,file="data20161019.Rda")
save(data20161020,file="data20161020.Rda")
save(data20161021,file="data20161021.Rda")
save(data20161022,file="data20161022.Rda")

for(i in 1:(nrow(OctDates2))){
  eval(parse(text=paste(datasets[i],"$CO2umol_mol<-
as.numeric(",datasets[i],"$CO2umol_mol)",sep = ' ')))
}
for(i in 1:(nrow(OctDates2))){
  eval(parse(text=paste(datasets[i],"$CH4umol_mol<-
as.numeric(",datasets[i],"$CH4umol_mol)",sep = ' ')))
}

format="%H:%M:%OS3"
op <- options(digits.secs=3)
data20161003$Time2<-strptime(gsub(":", ".", data20161003$Time),
format="%H.%M.%OS")
data20161003$Time2<-format(data20161003$Time2,format="%H:%M:%OS1")
data20161003$date<-strptime(paste(data20161003$Date, data20161003$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161003$date<-as.POSIXct(data20161003$date, tz="America/Denver")
save(data20161003,file="data20161003clean.Rda")

data20161004$Time2<-strptime(gsub(":", ".", data20161004$Time),
format="%H.%M.%OS")
data20161004$Time2<-format(data20161004$Time2,format="%H:%M:%OS1")
data20161004$date<-strptime(paste(data20161004$Date, data20161004$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161004$date<-as.POSIXct(data20161004$date, tz="America/Denver")

data20161005$Time2<-strptime(gsub(":", ".", data20161005$Time),
format="%H.%M.%OS")
data20161005$Time2<-format(data20161005$Time2,format="%H:%M:%OS1")
data20161005$date<-strptime(paste(data20161005$Date, data20161005$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161005$date<-as.POSIXct(data20161005$date, tz="America/Denver")

data20161006$Time2<-strptime(gsub(":", ".", data20161006$Time),
format="%H.%M.%OS")
data20161006$Time2<-format(data20161006$Time2,format="%H:%M:%OS1")
data20161006$date<-strptime(paste(data20161006$Date, data20161006$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161006$date<-as.POSIXct(data20161006$date, tz="America/Denver")

data20161007$Time2<-strptime(gsub(":", ".", data20161007$Time),
format="%H.%M.%OS")
data20161007$Time2<-format(data20161007$Time2,format="%H:%M:%OS1")

```

```

data20161007$date<-strptime(paste(data20161007$Date, data20161007$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161007$date<-as.POSIXct(data20161007$date, tz="America/Denver")

data20161010$Time2<-strptime(gsub(":", ".", data20161010$Time),
format="%H.%M.%OS")
data20161010$Time2<-format(data20161010$Time2,format="%H:%M:%OS1")
data20161010$date<-strptime(paste(data20161010$Date, data20161010$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161010$date<-as.POSIXct(data20161010$date, tz="America/Denver")

data20161011$Time2<-strptime(gsub(":", ".", data20161011$Time),
format="%H.%M.%OS")
data20161011$Time2<-format(data20161011$Time2,format="%H:%M:%OS1")
data20161011$date<-strptime(paste(data20161011$Date, data20161011$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161011$date<-as.POSIXct(data20161011$date, tz="America/Denver")

data20161012$Time2<-strptime(gsub(":", ".", data20161012$Time),
format="%H.%M.%OS")
data20161012$Time2<-format(data20161012$Time2,format="%H:%M:%OS1")
data20161012$date<-strptime(paste(data20161012$Date, data20161012$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161012$date<-as.POSIXct(data20161012$date, tz="America/Denver")

data20161013$Time2<-strptime(gsub(":", ".", data20161013$Time),
format="%H.%M.%OS")
data20161013$Time2<-format(data20161013$Time2,format="%H:%M:%OS1")
data20161013$date<-strptime(paste(data20161013$Date, data20161013$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161013$date<-as.POSIXct(data20161013$date, tz="America/Denver")

data20161014$Time2<-strptime(gsub(":", ".", data20161014$Time),
format="%H.%M.%OS")
data20161014$Time2<-format(data20161014$Time2,format="%H:%M:%OS1")
data20161014$date<-strptime(paste(data20161014$Date, data20161014$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161014$date<-as.POSIXct(data20161014$date, tz="America/Denver")

data20161017$Time2<-strptime(gsub(":", ".", data20161017$Time),
format="%H.%M.%OS")
data20161017$Time2<-format(data20161017$Time2,format="%H:%M:%OS1")
data20161017$date<-strptime(paste(data20161017$Date, data20161017$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161017$date<-as.POSIXct(data20161017$date, tz="America/Denver")
#save(data20161017,file="data20161017clean.Rda")

data20161018$Time2<-strptime(gsub(":", ".", data20161018$Time),
format="%H.%M.%OS")
data20161018$Time2<-format(data20161018$Time2,format="%H:%M:%OS1")
data20161018$date<-strptime(paste(data20161018$Date, data20161018$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161018$date<-as.POSIXct(data20161018$date, tz="America/Denver")
#save(data20161018,file="data20161018clean.Rda")

```

```

data20161019$Time2<-strptime(gsub(":", ".", data20161019$Time),
format="%H.%M.%OS")
data20161019$Time2<-format(data20161019$Time2,format="%H:%M:%OS1")
data20161019$date<-strptime(paste(data20161019$Date, data20161019$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161019$date<-as.POSIXct(data20161019$date, tz="America/Denver")
#save(data20161019,file="data20161019clean.Rda")

data20161020$Time2<-strptime(gsub(":", ".", data20161020$Time),
format="%H.%M.%OS")
data20161020$Time2<-format(data20161020$Time2,format="%H:%M:%OS1")
data20161020$date<-strptime(paste(data20161020$Date, data20161020$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161020$date<-as.POSIXct(data20161020$date, tz="America/Denver")
save(data20161020,file="data20161020clean.Rda")

data20161021$Time2<-strptime(gsub(":", ".", data20161021$Time),
format="%H.%M.%OS")
data20161021$Time2<-format(data20161021$Time2,format="%H:%M:%OS1")
data20161021$date<-strptime(paste(data20161021$Date, data20161021$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161021$date<-as.POSIXct(data20161021$date, tz="America/Denver")
save(data20161021,file="data20161021clean.Rda")

##Note: Need to get 10/24-10/31 values from Union10242016_11062016
rs = dbSendQuery(con, "SELECT DISTINCT Date FROM Union10242016_11062016 ORDER
BY Date")
OctDates<-fetch(rs, n=-1)
##Only want weekdays in october
OctDates3<-OctDates[c(1,2,3,4,5,8),]
OctDates3<-as.data.frame(OctDates3)
rm(OctDates)
datasets <- matrix(ncol=ncol(OctDates3), nrow=(nrow(OctDates3)))
for(i in 1:(nrow(OctDates3))){
  DBDate<-OctDates3[i,]
  sqlStatement <- paste("SELECT row_names, Date, Time,CO2umol_mol,
CH4umol_mol, wd, ws FROM Union10242016_11062016 WHERE Date='",DBDate, "'",
sep="")
  rs = dbSendQuery(con, sqlStatement)
  title<-gsub("-", "",paste("data", DBDate, sep = ''))
  datasets[i,]<-title
  assign(title, fetch(rs, n=-1))
  dbClearResult(rs)
}
for(i in 1:(nrow(OctDates3))){
  eval(parse(text=paste(datasets[i],"$CO2umol_mol<-
as.numeric(",datasets[i],"$CO2umol_mol)",sep = '')))
}
for(i in 1:(nrow(OctDates3))){
  eval(parse(text=paste(datasets[i],"$CH4umol_mol<-
as.numeric(",datasets[i],"$CH4umol_mol)",sep = '')))
}

data20161024$Time2<-strptime(gsub(":", ".", data20161024$Time),
format="%H.%M.%OS")

```

```

data20161024$Time2<-format(data20161024$Time2,format="%H:%M:%OS1")
data20161024$date<-strptime(paste(data20161024$Date, data20161024$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161024$date<-as.POSIXct(data20161024$date, tz="America/Denver")
#save(data20161024,file="data20161024clean.Rda")

data20161025$Time2<-strptime(gsub(":", ".", data20161025$Time),
format="%H.%M.%OS")
data20161025$Time2<-format(data20161025$Time2,format="%H:%M:%OS1")
data20161025$date<-strptime(paste(data20161025$Date, data20161025$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161025$date<-as.POSIXct(data20161025$date, tz="America/Denver")

data20161026$Time2<-strptime(gsub(":", ".", data20161026$Time),
format="%H.%M.%OS")
data20161026$Time2<-format(data20161026$Time2,format="%H:%M:%OS1")
data20161026$date<-strptime(paste(data20161026$Date, data20161026$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161026$date<-as.POSIXct(data20161026$date, tz="America/Denver")
head(data20161026)

data20161027$Time2<-strptime(gsub(":", ".", data20161027$Time),
format="%H.%M.%OS")
data20161027$Time2<-format(data20161027$Time2,format="%H:%M:%OS1")
data20161027$date<-strptime(paste(data20161027$Date, data20161027$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161027$date<-as.POSIXct(data20161027$date, tz="America/Denver")
head(data20161027)

data20161028$Time2<-strptime(gsub(":", ".", data20161028$Time),
format="%H.%M.%OS")
data20161028$Time2<-format(data20161028$Time2,format="%H:%M:%OS1")
data20161028$date<-strptime(paste(data20161028$Date, data20161028$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161028$date<-as.POSIXct(data20161028$date, tz="America/Denver")
head(data20161028)

data20161031$Time2<-strptime(gsub(":", ".", data20161031$Time),
format="%H.%M.%OS")
data20161031$Time2<-format(data20161031$Time2,format="%H:%M:%OS1")
data20161031$date<-strptime(paste(data20161031$Date, data20161031$Time2),
format="%Y-%m-%d %H:%M:%OS")
data20161031$date<-as.POSIXct(data20161031$date, tz="America/Denver")
head(data20161031)

# Check for NA values
which(is.na(data10$D))

#Need to amend 'datasets2' with 10/24-12/31
datasets3<-as.data.frame(datasets2)
datasets3<-as.character(datasets3$datasets)
datasets3<-append(datasets3, "data20161024")
datasets3<-append(datasets3, "data20161025")
datasets3<-append(datasets3, "data20161026")

```

```

datasets3<-append(datasets3, "data20161027")
datasets3<-append(datasets3, "data20161028")
datasets3<-append(datasets3, "data20161031")
datasets<-datasets3
OctDates<-as.character(OctDates2$OctDates2)
OctDates[16]<-"2016-10-24"
OctDates[17]<-"2016-10-25"
OctDates[18]<-"2016-10-26"
OctDates[19]<-"2016-10-27"
OctDates[20]<-"2016-10-28"
OctDates[21]<-"2016-10-31"

# summary data

FWU_stats_Date<- DB[,list(CO2mean=mean(CO2umol_mol),
  CH4mean=mean(CH4umol_mol),
  CO2min=min(CO2umol_mol),
  CH4min=min(CH4umol_mol),
  CH425=quantile(CH4umol_mol, .25, na.rm=TRUE),
  CO250=quantile(CO2umol_mol, .50, na.rm=TRUE),
  CH450=quantile(CH4umol_mol, .50, na.rm=TRUE),
  CO275=quantile(CO2umol_mol, .75, na.rm=TRUE),
  CH475=quantile(CH4umol_mol, .75, na.rm=TRUE),
  CO299=quantile(CO2umol_mol, .99, na.rm=TRUE),
  CH499=quantile(CH4umol_mol, .99, na.rm=TRUE),
  CO2max=max(CO2umol_mol),
  CH4max=max(CH4umol_mol)),
  by=format(date, "%Y/%m/%d")]

# Make boxplots of hourly CO2, CH4, WS
boxplot(data20161016$CO2umol_mol~ as.POSIXlt(data20161016$date)$hour,
  ylab="CO2 (umol/mol)",main="Hourly CO2 (umol/mol) 2016-10-16")
boxplot(data20161016$CH4umol_mol~ as.POSIXlt(data20161016$date)$hour,
  ylab="CH4 (umol/mol)",main="Hourly CH4 (umol/mol) 2016-10-16")
boxplot(data20161016$ws~ as.POSIXlt(data20161016$date)$hour, ylab="Wind Speed
(m/s)",main="Hourly Wind (m/s) 2016-10-16")

setwd("~/Dropbox (Personal)/Research/_ANALYSIS/UofU/Union/Oct/Images")
for(i in 1:length(datasets)){
  jpeg(file = paste("CO2Box_",datasets[i],".jpeg",collapse = '', sep = ''),
500, 340)
  boxplot(get(datasets[i])$CO2umol_mol~
as.POSIXlt(get(datasets[i])$date)$hour, ylab="CO2
(umol/mol)",main=paste("Hourly CO2 (umol/mol) ",OctDates[i],
sep=""),cex.lab=1.3,cex.axis=1.3, cex.main=1.5)
  dev.off()
}
for(i in 1:length(datasets)){
  jpeg(file = paste("CH4Box_",datasets[i],".jpeg",collapse = '', sep = ''),
500, 340)
  boxplot(get(datasets[i])$CH4umol_mol~
as.POSIXlt(get(datasets[i])$date)$hour, ylab="CH4
(umol/mol)",main=paste("Hourly CH4 (umol/mol) ",OctDates[i],
sep=""),cex.lab=1.3,cex.axis=1.3, cex.main=1.5)

```

```

    dev.off()
  }
  for(i in 1:length(datasets)){
    jpeg(file = paste("WSBox_",datasets[i],".jpeg",collapse = '', sep = ''),
    500, 340)
    boxplot(get(datasets[i])$ws~ as.POSIXlt(get(datasets[i])$date)$hour,
    ylab="Wind Speed (m/s)",main=paste("Hourly Wind Speed ",OctDates[i],
    sep=""),cex.lab=1.3,cex.axis=1.3, cex.main=1.5)
    dev.off()
  }
  boxplot(Oct$ws~ as.POSIXlt(Oct$date)$hour, ylab="Wind Speed
  (m/s)",main="Hourly Wind Speed Oct Weekdays",cex.lab=1.3,cex.axis=1.3,
  cex.main=1.5)
  boxplot(Oct$CH4umol_mol~ as.POSIXlt(Oct$date)$hour, ylab="CH4
  (umol/mol)",main="Hourly CH4 Oct Weekdays",cex.lab=1.3,cex.axis=1.3,
  cex.main=1.5)
  boxplot(Oct$CO2umol_mol~ as.POSIXlt(Oct$date)$hour, ylab="CO2
  (umol/mol)",main="Hourly CO2 Oct Weekdays",cex.lab=1.3,cex.axis=1.3,
  cex.main=1.5)

  maxCO2<-tail(sort(data20161021$CO2umol_mol),30)
  data20161021_2<-data20161021[data20161021$CO2umol_mol<1000,]

  data20161021_old<-data20161021
  data20161021<-data20161021_2
  rm(data20161021_old)

  ##---Removing duplicate time stamps
  data20161006<-subset(data20161006, !duplicated(date))
  for(i in 1:length(datasets)){
    assign(datasets[i],subset(get(datasets[i]), !duplicated(date)))
  }

  ###subestting wierd data
  quantile(Oct$CO2umol_mol, c(.0001,.9999))
  quantile(Oct$CO2umol_mol, c(.001,.999))
  tail(sort(Oct$CO2umol_mol),100)
  HighCO2<-subset(Oct, Oct$CO2umol_mol>5000)
  unique(HighCO2$date) #"2016-10-05" "2016-10-17" "2016-10-21" "2016-10-31"
  #Oct 5th, 27th, 21
  Oct5_TS<-cbind(data20161005$CO2umol_mol, data20161005$CH4umol_mol)
  Oct5_TS<-zoo(Oct5_TS, data20161005$date)
  plot(Oct5_TS, plot.type="multiple", main="Union Licor 10/5/16",xlab="",
  ylab=list("CO2", "CH4(umol/mol)"),col=3:4)
  Oct17_TS<-cbind(data20161017$CO2umol_mol, data20161017$CH4umol_mol)
  Oct17_TS<-zoo(Oct17_TS, data20161017$date)
  plot(Oct17_TS, plot.type="multiple", main="Union Licor 10/17/16",xlab="",
  ylab=list("CO2", "CH4(umol/mol)"),col=3:4)
  Oct21_TS<-cbind(data20161021$CO2umol_mol, data20161021$CH4umol_mol)
  Oct21_TS<-zoo(Oct21_TS, data20161021$date)
  plot(Oct21_TS, plot.type="multiple", main="Union Licor 10/21/16",xlab="",
  ylab=list("CO2", "CH4(umol/mol)"),col=3:4)
  Oct31_TS<-cbind(data20161031$CO2umol_mol, data20161031$CH4umol_mol)

```

```

Oct31_TS<-zoo(Oct31_TS, data20161031$date)
plot(Oct31_TS, plot.type="multiple", main="Union Licor 10/31/16",xlab="",
ylab=list("CO2", "CH4(umol/mol)"),col=3:4)
#decided to remove CO2 values > 32010
Oct_test<-subset(Oct, Oct$CO2umol_mol<32011)
Oct_test2<- subset(Oct_test, Oct_test$CO2umol_mol>300)
Oct_unclean<-Oct
rm(Oct)
Oct<-Oct_test2
rm(Oct_test)
rm(Oct_test2)
Oct$Hour<-hour(Oct$date)

# Make boxplots of hourly CO2, CH4, WS
boxplot(Oct$CO2umol_mol~ as.POSIXlt(Oct$date)$hour, ylab="CO2
(umol/mol)",main="Weekday Hourly CO2 (umol/mol)")
boxplot(Oct$CH4umol_mol~ as.POSIXlt(Oct$date)$hour, ylab="CH4
(umol/mol)",main="Weekday Hourly CH4 (umol/mol)")
boxplot(Oct$ws~ as.POSIXlt(Oct$date)$hour, ylab="Wind Speed
(m/s)",main="Weekday Hourly Wind Speed (m/s)")

setwd("~/Dropbox (Personal)/Research/_ANALYSIS/UofU/Union/Oct/Images")
for(i in 1:length(datasets)){
  title<-paste(datasets[i],"_TS",sep = '')
  assign(title, cbind(get(datasets[i])$CO2umol_mol,
get(datasets[i])$CH4umol_mol))
  assign(title, zoo(get(title), get(datasets[i])$date))
  jpeg(file = paste("TS_",datasets[i],"Unclean.jpeg",collapse = '', sep = ''),
500, 340)
  plot(get(title), plot.type="multiple", main=paste("Union Licor",OctDates[i],
sep=""),xlab="", ylab=list("CO2(umol/mol)", "CH4(umol/mol)"),col=3:4)
  dev.off()
}

setwd("~/Dropbox (Personal)/Research/_ANALYSIS/UofU/Union/Oct/Images")
for(i in 1:length(datasets)){
  jpeg(file = paste("CO2Box_",datasets[i],".jpeg",collapse = '', sep = ''),
500, 340)
  boxplot(get(datasets[i])$CO2umol_mol~
as.POSIXlt(get(datasets[i])$date)$hour, ylab="CO2
(umol/mol)",main=paste("Hourly CO2 (umol/mol) ",OctDates[i],
sep=""),cex.lab=1.3,cex.axis=1.3, cex.main=1.5)
  dev.off()
}

which(Oct$CO2umol_mol==53908700)
Oct[18973314,]
### note it looks like there are a lot of duplicates in Oct

## 10/17, 10/21

weird20161018<-subset(data20161018, date >= as.POSIXct('2016-10-18 22:00') &
date <=

```

```

as.POSIXct('2016-10-18 22:59'))

weird20161017<-subset(data20161017, date >= as.POSIXct('2016-10-17 06:00') &
date <=
as.POSIXct('2016-10-17 07:59'))

data20161016_2<-subset(data20161016,data20161016$CO2umol_mol<3000 &
data20161016$CO2umol_mol>300)
boxplot(data20161016_2$CO2umol_mol~ as.POSIXlt(data20161016_2$date)$hour,
ylab="CO2 (umol/mol)",main="Hourly CO2 (umol/mol) 2016-10-16")
data20161016<-data20161016_2
rm(data20161016_2)

data20161017_2<-subset(data20161017,data20161017$CO2umol_mol<700 &
data20161017$CO2umol_mol>350)
boxplot(data20161017_2$CO2umol_mol~ as.POSIXlt(data20161017_2$date)$hour,
ylab="CO2 (umol/mol)",main="Hourly CO2 (umol/mol) 2016-10-17")
data20161017<-data20161017_2
rm(data20161017_2)

data20161019_2<-subset(data20161019,data20161019$CO2umol_mol<1000 &
data20161019$CO2umol_mol>300)
boxplot(data20161019_2$CO2umol_mol~ as.POSIXlt(data20161019_2$date)$hour,
ylab="CO2 (umol/mol)",main="Hourly CO2 (umol/mol) 2016-10-19")
data20161019<-data20161019_2
rm(data20161019_2)

data20161021_2<-subset(data20161021,data20161021$CO2umol_mol<700 &
data20161021$CO2umol_mol>375)
boxplot(data20161021_2$CO2umol_mol~ as.POSIXlt(data20161021_2$date)$hour,
ylab="CO2 (umol/mol)",main="Hourly CO2 (umol/mol) 2016-10-21")
data20161021<-data20161021_2
rm(data20161021_2)

data20161020_2<-subset(data20161020, data20161020$CH4umol_mol>1)
boxplot(data20161020_2$CH4umol_mol~ as.POSIXlt(data20161020_2$date)$hour,
ylab="CH4 (umol/mol)",main="Hourly CH4 (umol/mol) 2016-10-20")

## ===== Timeseries
library(xts)
zoo20161016<-zoo(data20161016$CO2umol_mol,data20161016$date) #use zoo becuase
they are not constant time intervals
plot(zoo20161016, main = "CO2 2016-10-16", ylab="CO2 (umol/mol)", xlab=NULL) #
this needs some formatting
abline(h=quantile(data20161016$CO2umol_mol, c(.99)), col="red", lwd=2) #adds
99th percentile

### Timeseries of the means of hourly
#merging the week of data into one:
Oct1<-
rbind(data20161003,data20161004,data20161005,data20161006,data20161007,data201
61010,data20161011,data20161012,data20161013,data20161014)
Oct2<-
rbind(data20161017,data20161018,data20161019,data20161020,data20161021,data201
61024,data20161025,data20161026,data20161027,data20161028,data20161031)

```



```

Oct<-rbind(Oct1, Oct2)
setwd("~/Dropbox (Personal)/Research/_ANALYSIS/UofU/Union/Oct")
save(Oct,file="OctoberWeekDay.Rda")
summary(Oct$CO2umol_mol)

means <-aggregate(Oct2$CO2umol_mol, list(format(Oct2$date, "%w-%H")),
mean,na.rm = TRUE)
plot(means$x, xaxt = "n", type = "n",
      xlab = "day of week",
      ylab = "CO2 (umol/mol)",
      main = "CO2 at Union Oct16-23")
axis(1, at = seq(1, 169, 24), labels = FALSE)
days = c("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")
loc.days = seq(13, 157, 24) # location of labels on x-axis
mtext(days, side = 1, line = 1, at = loc.days)
abline(v = seq(1, 169, 24), col = "grey85")
lines(means$x, col = "darkorange2", lwd = 2)

means2 <-aggregate(Oct2$CH4umol_mol, list(format(Oct2$date, "%w-%H")),
mean,na.rm = TRUE)
plot(means2$x, xaxt = "n", type = "n",
      xlab = "day of week",
      ylab = "CH4 (umol/mol)",
      main = "CH4 at Union Oct16-23")
axis(1, at = seq(1, 169, 24), labels = FALSE)
days = c("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")
loc.days = seq(13, 157, 24) # location of labels on x-axis
mtext(days, side = 1, line = 1, at = loc.days)
abline(v = seq(1, 169, 24), col = "grey85")
lines(means2$x, col = "aquamarine4", lwd = 2)

## Histograms
hist(Oct2$ws, col="purple", main="Wind Speed Frequency", xlab="Wind Speed
(m/s)")

##Wind rose
windRose(Oct, main="UofU Wind Rose Oct 2016")

# wind roses by day
sunrise<-parse_date_time("8:00:00", orders="hms")
sunset<-parse_date_time("19:00:00", orders="hms")
for(i in 2:length(datasets)){
  get(datasets[i])$daylight<-ifelse(
    hour(get(datasets[i])$date)>=hour(sunrise) &
    hour(get(datasets[i])$date)<hour(sunset), "daylight", "nighttime")
  }
Oct2$daylight<-ifelse(
  hour(Oct2$date)>=hour(sunrise) & hour(Oct2$date)<hour(sunset), "daylight",
  "nighttime")
windRose(Oct2, daylight= "nighttime", main="Night UofU Wind Rose Oct 16-23,
2016")
windRose(Oct2, daylight= "daylight", main="Daytime UofU Wind Rose Oct 16-23,
2016")

```

```

windRose(Oct2, type="daylight", main="UofU Wind Rose Oct 16-23, 2016")

#now make a loop for open air with the datasets table
for(i in 1:length(datasets)){
  windRose(get(datasets[i]), main=paste("UofU Wind Rose ",OctDates[i],
sep=""))
  polarPlot(get(datasets[i]), pollutant="CH4umol_mol", statistic="cpf",
percentile=c(99,100), min.bin=3, main=paste("UofU 99% CBPF CH4 ",OctDates[i],
sep=""))
  polarPlot(get(datasets[i]), pollutant="CO2umol_mol", statistic="cpf",
percentile=c(99,100), min.bin=3, main=paste("UofU 99% CBPF CO2 ",OctDates[i],
sep=""))
}
##Subest data by hour for specific CBPF
Hours<-unique(Oct$Hour)
which(Oct$Hour=="NA")
for(i in 1:length(Hours)){
  title<-paste("Hour", Hours[i], sep = '')
  assign(title, Oct2[Oct2$Hour==Hours[i],])
}
polarPlot(Oct, pollutant="CH4umol_mol", statistic="cpf", percentile=c(99,100),
min.bin=3, main=paste("UofU 99% CBPF CH4 Weekdays October 2016"))
polarPlot(Oct, pollutant="CO2umol_mol", statistic="cpf", percentile=c(99,100),
min.bin=3, main=paste("UofU 99% CBPF CO2 Weekdays October 2016"))
windRose(Oct, main="UofU Wind Rose at hour Weekdays October 2016")
setwd("~/Dropbox
(Personal)/Research/_ANALYSIS/UofU/Union/Oct/Images/Thesis_images")
for(i in 6:17){
  jpeg(file = paste("CH4_99_CBPF_",Hours[i],".jpeg",collapse = '', sep = ''),
500, 340)
  polarPlot(subset(Oct,Hour==i), pollutant="CH4umol_mol", statistic="cpf",
percentile=c(99,100), min.bin=3, main=paste("UofU 99% Weekday CH4 at hour
",Hours[i], sep=""))
  dev.off()
}
for(i in 6:17){
  jpeg(file = paste("CO2_99_CBPF_",Hours[i],".jpeg",collapse = '', sep = ''),
500, 340)
  polarPlot(subset(Oct,Hour==i), pollutant="CO2umol_mol", statistic="cpf",
percentile=c(99,100), min.bin=3, main=paste("UofU 99% Weekday CO2 at hour
",Hours[i], sep=""))
  dev.off()
}
for(i in 6:17){
  jpeg(file = paste("Windrose_",Hours[i],".jpeg",collapse = '', sep = ''),
500, 340)
  windRose(subset(Oct,Hour==i), main=paste("UofU Weekday Wind Rose at hour
",Hours[i], sep=""))
  dev.off()
}
for(i in 6:17){
  polarPlot(Oct, pollutant="CH4umol_mol", statistic="cpf",
percentile=c(99,100), min.bin=3, main=paste("UofU 99% CBPF CH4 at hour

```

```

",Hours[i], sep=""))
  polarPlot(Oct, pollutant="CO2umol_mol", statistic="cpf",
percentile=c(99,100), min.bin=3, main=paste("UofU 99% CBPF CO2 at hour
",Hours[i], sep=""))
  windRose(Oct, main=paste("UofU Wind Rose at hour ",Hours[i], sep=""))
}
for(i in 1:length(datasets)){
  percentileRose(get(datasets[i]), pollutant="CH4umol_mol", percentile=99,
method="cpf", col="darkorange", smooth=TRUE, main=paste("UofU 99% CH4 CPF
",OctDates[i], sep=""))
  percentileRose(get(datasets[i]), pollutant="CO2umol_mol", percentile=99,
method="cpf", col="lightblue", smooth=TRUE, main=paste("UofU 99% CO2 CPF
",OctDates[i], sep=""))
}

polarPlot(Oct, pollutant="CH4umol_mol", statistic="cpf", percentile=c(99,100),
min.bin=3, main="UofU 99% Weekday CH4")
polarPlot(Oct, pollutant="CH4umol_mol", statistic="cpf", percentile=c(95,99),
min.bin=3, main="UofU 95-99% Weekday CH4")
polarPlot(Oct, pollutant="CH4umol_mol", statistic="cpf", percentile=c(90,95),
min.bin=3, main="UofU 90-95% Weekday CH4")
polarPlot(Oct, pollutant="CH4umol_mol", statistic="cpf", percentile=c(75,90),
min.bin=3, main="UofU 75-90% Weekday CH4")
polarPlot(Oct, pollutant="CH4umol_mol", statistic="cpf", percentile=c(50,75),
min.bin=3, main="UofU 50-75% Weekday CH4")

polarPlot(Oct, pollutant="CO2umol_mol", statistic="cpf", percentile=c(99,100),
min.bin=3, main="UofU 99% Weekday CO2")
polarPlot(Oct, pollutant="CO2umol_mol", statistic="cpf", percentile=c(95,99),
min.bin=3, main="UofU 95-99% Weekday CO2")
polarPlot(Oct, pollutant="CO2umol_mol", statistic="cpf", percentile=c(90,95),
min.bin=3, main="UofU 90-95% Weekday CO2")
polarPlot(Oct, pollutant="CO2umol_mol", statistic="cpf", percentile=c(75,90),
min.bin=3, main="UofU 75-90% Weekday CO2")
polarPlot(Oct, pollutant="CO2umol_mol", statistic="cpf", percentile=c(50,75),
min.bin=3, main="UofU 50-75% Weekday CO2")

windRose(Oct, main="UofU Wind Rose (entire)")

polarPlot(get(paste("Hour",i,sep='')), pollutant="CO2umol_mol",
statistic="cpf", percentile=c(99,100), min.bin=3, main=paste("UofU 99% CBPF
CO2 at hour ",Hours[i], sep=""))
windRose(get(paste("Hour",i,sep='')), main=paste("UofU Wind Rose at hour
",Hours[i], sep=""))

percentileRose(DB, pollutant="CH4_dry_sync", method="cpf",

###-----Covariance
cor(Oct[,4:7], method="spearman")
cor(Oct, 'CO2umol_mol', 'CH4umol_mol', hour(Oct$date), method="spearman")
Oct$Hour<-hour(Oct$date)
#Na's with Hour
which(is.na(Oct$Hour))
Oct2<-Oct[-c(2491734, 3355735), ]

```

```

Oct<-Oct2
rm(Oct2)
Cor_sp<-cor(Oct[,c(4,5,6,7,11)], method="spearman")
write.table(Cor_sp, file="spear.csv",sep=",")
Cor_pear<-cor(Oct[,c(4,5,6,7,11)])
cov(Oct2[,4:7])
## Covariance by windspeed
Oct$ws_group<-as.integer(Oct$ws)
for(i in 0:max(Oct$ws_group)){
  data<-subset(Oct,ws_group==i)
  title<-paste("Cor",i,sep='')
  assign(title, cor(data[,c(4,5,6,7,11)],method="spearman"))
  write.table(get(title), file=paste("Cor",i,".csv",sep=''),sep=",")
}

for(Oct2$ws_group==4){
  histo(Oct2$CO2umol_mol)
}
#boxplot of enitre data by windspeed
boxplot(Oct2$CO2umol_mol~ Oct$ws, ylab="CO2 (umol/mol)",main="CO2 (umol/mol)
by windspeed")

#Stats for weekday
## something is wrong maybe make it as data.frame
OctWeekdayStats<- Oct[,list(CO2mean=mean(CO2umol_mol),
                           CH4mean=mean(CH4umol_mol),
                           CO2min=min(CO2umol_mol),
                           CH4min=min(CH4umol_mol),
                           CO225=quantile(CO2umol_mol, .25, na.rm=TRUE),
                           CH425=quantile(CH4umol_mol, .25, na.rm=TRUE),
                           CO250=quantile(CO2umol_mol, .50, na.rm=TRUE),
                           CH450=quantile(CH4umol_mol, .50, na.rm=TRUE),
                           CO275=quantile(CO2umol_mol, .75, na.rm=TRUE),
                           CH475=quantile(CH4umol_mol, .75, na.rm=TRUE),
                           CO299=quantile(CO2umol_mol, .99, na.rm=TRUE),
                           CH499=quantile(CH4umol_mol, .99, na.rm=TRUE),
                           CO2max=max(CO2umol_mol),
                           CH4max=max(CH4umol_mol),
                           CO2SD=sd(CO2umol_mol),
                           CH4SD=sd(CH4umol_mol)),
                          by=format(date, "%Y-%m-%d")]
OctWeekdayStats<- Oct[,list(CO2mean=mean(CO2umol_mol),
                           CH4mean=mean(CH4umol_mol),
                           CO2min=min(CO2umol_mol),
                           CH4min=min(CH4umol_mol),
                           CO225=quantile(CO2umol_mol, .25, na.rm=TRUE),
                           CH425=quantile(CH4umol_mol, .25, na.rm=TRUE),
                           CO250=quantile(CO2umol_mol, .50, na.rm=TRUE),
                           CH450=quantile(CH4umol_mol, .50, na.rm=TRUE),
                           CO275=quantile(CO2umol_mol, .75, na.rm=TRUE),
                           CH475=quantile(CH4umol_mol, .75, na.rm=TRUE),
                           CO299=quantile(CO2umol_mol, .99, na.rm=TRUE),
                           CH499=quantile(CH4umol_mol, .99, na.rm=TRUE),

```

```

        CO2max=max(CO2umol_mol),
        CH4max=max(CH4umol_mol),
        CO2SD=sd(CO2umol_mol),
        CH4SD=sd(CH4umol_mol)),
        by=list(date)]
OctWeekdayStats<-Oct[,list(CO2mean=mean(CO2umol_mol),
        CH4mean=mean(CH4umol_mol),
        CO2min=min(CO2umol_mol),
        CH4min=min(CH4umol_mol),
        CO225=quantile(CO2umol_mol, .25, na.rm=TRUE),
        CH425=quantile(CH4umol_mol, .25, na.rm=TRUE),
        CO250=quantile(CO2umol_mol, .50, na.rm=TRUE),
        CH450=quantile(CH4umol_mol, .50, na.rm=TRUE),
        CO275=quantile(CO2umol_mol, .75, na.rm=TRUE),
        CH475=quantile(CH4umol_mol, .75, na.rm=TRUE),
        CO299=quantile(CO2umol_mol, .99, na.rm=TRUE),
        CH499=quantile(CH4umol_mol, .99, na.rm=TRUE),
        CO2max=max(CO2umol_mol),
        CH4max=max(CH4umol_mol),
        CO2SD=sd(CO2umol_mol),
        CH4SD=sd(CH4umol_mol)),
        by=list(date, format="%Y-%m-%d"))]
Test1<-Oct[,list(CO2mean=mean(CO2umol_mol),
        CH4mean=mean(CH4umol_mol),
        CO2min=min(CO2umol_mol),
        CH4min=min(CH4umol_mol)),
        ]

# Combine Weekday and weekend into one
save(Oct,file="WeekdayOct_Unclean.Rda")
save(Oct2,file="WeekendOct_Unclean.Rda")

#data.table summary statistics - need to combine weekend weekday
#FWU_stats_Date<- DB[,list(CO2drymean=mean(CO2_dry_sync),
#
#         CH4drymean=mean(CH4_dry_sync),
#
#         CO2drymin=min(CO2_dry_sync),
#
#         CH4drymin=min(CH4_dry_sync),
#
#         CO2dry25=quantile(CO2_dry_sync, .25, na.rm=TRUE),
#
#         CH4dry25=quantile(CH4_dry_sync, .25, na.rm=TRUE),
#
#         CO2dry50=quantile(CO2_dry_sync, .50, na.rm=TRUE),
#
#         CH4dry50=quantile(CH4_dry_sync, .50, na.rm=TRUE),
#
#         CO2max=max(CO2_dry_sync),
#
#         CH4max=max(CH4_dry_sync)),
#
#         by=format(date, "%Y/%m/%d")]

##Two-sample t-test
# http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R4\_One-TwoSampleTests-ANOVA/R4\_One-TwoSampleTests-ANOVA\_print.html

```

REFERENCES

- Ashbaugh, L.; Malm, W.; Sadeh, W. A Residence Time Probability Analysis of Sulfur Concentrations at Grand Canyon National Park. *Atmos. Environ.* **1985**, *19* (8), 1263–1270.
- Carslaw, D. *The Openair Manual. Open-Source Tools for Analysing Air Pollution Data*; 2015.
- Cosemans, G.; Kretzschmar, J.; Mensink, C. Pollutant Roses for Daily Averaged Ambient Air Pollutant Concentrations. *Atmos. Environ.* **2008**, *42* (29), 6982–6991.
- Fisher, N. *Statistical Analysis of Circular Data*; Cambridge University Press: Cambridge, UK, 1996.
- Kim, E.; Hopke, P. K. Comparison between Conditional Probability Function and Nonparametric Regression for Fine Particle Source Directions. *Atmos. Environ.* **2004**, *38* (28), 4667–4673.
- Krecl, P.; Targino, A. C.; Wiese, L.; Ketzel, M.; de Paula Correa, M. Screening of Short-Lived Climate Pollutants in a Street Canyon in a Mid-Sized City in Brazil. *Atmos. Pollut. Res.* **2016**, *7* (6), 1022–1036.
- Lewicki, J. L.; Hilley, G. E.; Fischer, M. L.; Pana, L.; Oldenburg, C. M.; Dobeck, L.; Spangler, L. Detection of CO₂ Leakage by Eddy Covariance during the ZERT Project's CO₂ Release Experiments. *Energy Procedia* **2009**, *1* (1), 2301–2306.
- Minck, L. K. Design, Deployment, and Initial Calibration of a Tower for Greenhouse Gas Measurements, University of Utah, 2015.
- Pekney, N. J.; Davidson, C. I.; Zhou, L.; Hopke, P. K. Application of PSCF and CPF to PMF-Modeled Sources of PM_{2.5} in Pittsburgh. *Aerosol Sci. Technol.* **2006**, *40* (10), 952–961.
- Ruiz, S.; Fernández-Olmo, I.; Irabien, Á. Discussion on Graphical Methods to Identify Point Sources from Wind and Particulate Matter-Bound Metal Data. *Urban Clim.* **2014**, *10*, 671–681.
- Uria-Tellaetxe, I.; Carslaw, D. C. Conditional Bivariate Probability Function for Source Identification. *Environ. Model. Softw.* **2014**, *59*, 1–9.

Varland, A. Engineering Design and Assembly of a Surface Methane and Carbon Dioxide Gas Flux Measurement Chamber with a Case Study on the University of Utah Campus (Unpublished Master's Thesis), University of Utah, 2014.

Xie, Y.; Berkowitz, C. The Use of Conditional Probability Functions and Potential Source Contribution Functions to Identify Source Regions and Advection Pathways of Hydrocarbon Emissions in Houston, TX. *Atmos. Environ.* **2007**, *41* (28), 5831–5847.

Yang, Y.-M.; Small, M. J.; Ogretim, E. O.; Gray, D. D.; Bromhal, G. S.; Strazisar, B. R.; Wells, A. W. Probabilistic Design of a Near-Surface CO₂ Leak Detection System. *Environ. Sci. Technol.* **2011**, *45* (15), 6380–6387.

Zardi, D.; Whiteman, C. Diurnal Mountain Wind Systems. *Mt. Weather Res. Forecast.* **2013**, 35–119.